Journal of Intelligent Systems: Theory and Applications

JISTA

Journal of Intelligent Systems: Theory and Applications

SEPTEMBER 2024

ISSN: 2651-3927

VOL 7 NO 2

ARTIFICIAL INTELLIGENT > MACHINE LEARNING > DEEP LEARNING

https://dergipark.org.tr/en/pub/jista

Journal of Intelligent Systems:
Theory and Applications
2024 Volume: 7 Issue: 2

Editorial Boards

Journal of Intelligent Systems:
Theory and Applications
2024 Volume: 7 Issue: 2

Language Editor

Editorial Advisory Board

Editorial Assistants

https://dergipark.org.tr/en/pub/jista

Journal of Intelligent Systems:
Theory and Applications
2024 Volume: 7 Issue: 2

Contents

Research Articles

Research Article
Journal of Intelligent Systems: Theory and Applications 7(2) (2024) 43-55
DOI: 10.38016/jista.1200225

Assessment of effective factors on student performance based on machine learning methods

Hasan Yıldırım[1*]

[1] Department of Mathematics, Karamanoğlu Mehmetbey University, Karaman, Türkiye

hasanyildirim@kmu.edu.tr

Abstract

Machine learning methods have gained increasing attention in the field of education due to advancing technological tools and rapidly growing data. The general focus of this attention is on identifying the best method, but it is also critical to determine the extent to which the methods under consideration differ statistically and to correctly identify variable importance metrics. In this study, we benchmarked the performance of twenty-three machine learning algorithms on real educational data via cross-validation based on criteria such as accuracy, AUC and F1-score. Besides, the methods were statistically compared using DeLong and McNemar tests. The findings showed that the LightGBM method appeared to be the best method and presented the most important factors determining student achievement according to this method. The systematic process followed in the study is considered to yield valuable insights for data-driven studies as well as the field of education.

Keywords: Student performance, Machine learning, Artificial intelligence, Feature selection, Statistical analysis

1. Introduction

Artificial intelligence is increasingly deeply integrated into real life and is enhancing human beings' ability to predict routines, capacities, and behaviors. With the technological facilities that are being developed to achieve these goals, the size of the data collected is also expanding proportionally to the ability to collect and process data. The field of education is arguably one of the fields generating the highest amount of valuable knowledge from the data gathered. Artificial intelligence models can be effectively used for primary purposes such as students' performance evaluations (in-term and end-of-term), dropout status, and identification of individuals at risk (Albreiki et al., 2021). The beneficial results provided by artificial intelligence models have expanded their usages in education and enabled them to prepare personalized (Li et al., 2020), updated (Guan et al., 2020) course content, developed effective course selection tools (Tilahun ve Sekeroglu, 2020), and prepared exam formats (Wu et al., 2020).

Educational research is conducted not only to improve decisions locally, but also on the results of several examinations (such as program for international student assessment (PISA), trends in international mathematics and science study (TIMSS), etc.) carried out globally. The motivation underlying these studies is to improve the socio-economic status and quality of life of both individuals and the countries in which they live through improving the quality of education (Sağlam and Aydoğmuş, 2016). Machine learning, as the most important sub-field of artificial intelligence, contributes significantly to realizing this motivation and providing accurate recommendations to decision (policy) makers.

The usage of machine learning models in the field of education is particularly focused on the supervised learning. Supervised learning is based on assuming that the quantitative (e.g., student grade: 82/100, student attendance percentage: 73%) or qualitative (e.g., student achievement status: failed or success, student grade: AA or FF) variable that is the focus of the study is known and accurately predicted by a set of variables that are expected to affect it. The most widely used algorithms in the literature for this type of learning (Sekeroglu et al., 2021) are logistic regression (LR), naive bayes (NB), k-nearest neighbor (KNN), classification and regression trees (CART), linear regression (LIN), random forests (RF), bagging (BG), gradient boosting machine (GBM), extreme gradient boosting (Xgboost), artificial neural networks (ANN), support vector machines (SVMs), extreme learning machine (ELM), long short-term memory (LSTM), deep neural networks (DNN). In the literature, there are numerous studies involving such

* Corresponding Author
E-mail: hasanyildirim@kmu.edu.tr

Recieved : 31 Oct 2023
Revision : 14 Feb 2024
Accepted : 2 Jul 2024

models, some of the prominent studies can be given as follows:

Gamuling et al. (2016) studied student performance prediction in blended learning environments using discrete Fourier transforms (DFT) and various machine learning methods (including KNN, SVM, ANN and NB). Elbadrawy et al. (2016) proposed to utilize random forests, personalized multi-regression, and matrix factorization approaches to predict students' grades and assessments in future courses. Tran et al. (2017) proposed a unified system that connects classical machine learning methods (LR, CART and SVM) and recommender systems to predict student performance.) Like Tran et al. (2017), but not including the outputs of recommender systems, Adejo and Connolly (2018) presented an ensemble model incorporating cart, ann and svm methods to predict student performance. Hussain et al. (2019) employed various methods such as CART, LR, ANN, SVM and NB to identify the difficulties encountered by students during the term and to improve their performance. Yousafzai et al. (2020) employed genetic algorithm, CART and KNN models through both classification and regression models to predict student performance. Deo et al. (2020) have proposed models such as ELM, RF and Volterra to predict student performance in engineering mathematics courses and presented the results comparatively. Assellman et al. (2021) utilized RF and some boosting-based algorithms (including Adaboost and Xgboost) to accurately predict student performance. Suleiman and Anane (2022) have applied LR, CART, SVM and RF algorithms to predict the cumulative grade of students based on their performance in different years. Pallathadka et al. (2023) have comparatively presented the results of Naive Bayes, ID3, C4.5, and SVM models for predicting student performance. Chen and Zhai (2023) have compared the results of KNN, CART, RF, LR, SVM, NB, and ANN models in different application scenarios using several different datasets. Extensive studies on this topic are currently ongoing and comprehensive listings of these studies categorized according to aims, methods and outcomes can be obtained from the reviews by Albreiki et al. (2021), Sekeroglu et al. (2021) and Alalawi et al. (2023).

1.1. Study Aims and Motivation

The use of machine learning models in the literature is beneficial to a certain extent, however, some aspects have been relatively often disregarded:

i. It is critical in data-driven education studies to realize this motivation by not only estimating the value of the target variable that is the focus of the study, but also identifying the important factors that affect it. The variable importance measures can lead to more compact and scalable models.

ii. The statistical significance in performance comparisons of machine learning models can provide additional insights in model selection. The principle that the best model is the simplest model can be followed unless there is a significant difference.

This study focuses on these two mentioned perspectives and presents a comprehensive comparison of best machine learning algorithms. The content of the study is summarized as follows: The methods evaluated in the study are given in Section 2. Section 3 provides details about the experimental process. Model training results are presented in Section 4. Finally, the discussion and summary comments on the results of the study are reported in Section 5.

Table 1. List of models (algorithms) evaluated in the study

Type	Abbrevation	Model (Authors)	Brief Explanation
Instance-based	KNN	K-nearest neigbors (Cover and Hart, 1967)	The versatile algorithm employed in machine learning for both classification and regression tasks, and operates on the principle that similar data points are generally close in feature space. Its applications range from recommender systems to pattern recognition and anomaly detection, making it invaluable in academic and industrial contexts.
Statistical	NB	Naive bayes (Domingos and Pazzani, 1997)	The probabilistic machine learning algorithm based on Bayes' Theorem, which assumes strong (naive) independence between features. It is particularly effective for classification tasks including spam detection and sentiment analysis due to its simplicity, efficiency and ability to handle large datasets.
	LR	Logistic regression (Cox, 1958)	The statistical model widely utilized for binary classification tasks, such as predicting whether an event will occur or not. It estimates probabilities using a logistic function, making it ideal for scenarios where outcomes are categorical and decisions are probabilistic.
	PLS	Partial least squares (Wold,	An extension of the partial least squares algorithm that particularly addresses the prediction of continuous dependent variables is partial least squares regression. By finding the directions of

		1982; Wold et al., 1984)	greatest variance that closely relate independent variables to the dependent variable, it constructs predictive models. It is particularly effective when there are multicollinearity problem in data set or high dimensional settings. Therefore, it is highly applicable in both research and practical problem solving.
Tree and rule-based	CART	Classification and Regression Trees (Breiman et al., 1984)	A nonparametric decision tree learning technique that is suitable for both classification and regression tasks. It forms binary trees by partitioning the dataset into subsets based on feature values maximizing the separation of data in terms of the purity of the target variable. The CART is notorious for its interpretability and flexibility, which makes it a practical solution and a popular choice in areas where clear decision rules are required.
	C5.0	C5.0 (Quinlan, 1992; Quinlan, 1993)	An advanced decision tree algorithm that builds on predecessors like ID3 and C4.5, enhancing accuracy through boosting, winnowing, and pruning. It is widely used for classification and adapted for regression, excelling in handling large datasets and is popular for its robust performance and interpretability.
	C5.0-Rules	C5.0-rules (Quinlan, 1992; Quinlan, 1993)	C5.0-rules is a variation of the C5.0 algorithm that generates a set of decision rules rather than a tree structure, tailored for classification and adaptable for regression tasks. This approach simplifies the decision-making process by extracting the most significant rules from data, enhancing interpretability and accuracy.
	RuleFit	RuleFit (Friedman and Popescu, 2008)	A machine learning algorithm that combines decision tree-like rules with linear regression models to predict outcomes. It generates rules from an ensemble of trees and uses them as features in a linear model, effectively capturing both linear and interaction effects among variables. It is particularly valued for its interpretability and precision, making it suitable for applications in fields like healthcare and finance where understanding the model's decision process is crucial.
	BAT	Bayesian additive trees (Chipman et al., 2010)	A statistical model that uses Bayesian methods to combine multiple decision tree models for more reliable predictions. It estimates complex functions by averaging over many trees, improving accuracy and robustness while providing credible intervals for predictions. It is particularly effective in scenarios requiring careful uncertainty estimation, such as in medical prognosis and economic forecasting.
Neural network-based	MLP	Multilayer perceptron (Hornik et al., 1989)	A form of deep learning where an MLP, a type of artificial neural network, is used to classify data into distinct categories. It features multiple layers of neurons with non-linear activation functions, enabling it to capture complex patterns and relationships in data.
Spline-based	MARS	Multivariate adaptive regression spline (Friedman, 1991)	A non-parametric technique that models relationships within data by fitting piecewise linear splines, which are flexible enough to capture complex patterns. It's particularly useful in scenarios where the relationship between variables is non-linear and intricate, adjusting automatically to changes in data trends.
Kernel-based	SVM	Support Vector Machines (Vapnik et al., 1996; Schölkopf and Smola, 2002)	A powerful class of supervised learning models used for classification and regression tasks. They work by finding the hyperplane that best separates different classes in the feature space, maximizing the margin between data points of different categories. This capability to handle both linear and non-linear boundaries makes SVMs highly effective in diverse applications such as image recognition, bioinformatics, and text categorization.
Ensembles	Bag	Bagging (Breiman, 1996)	An ensemble machine learning technique used to improve the stability and accuracy of classification algorithms. It involves creating multiple versions of a predictor model by training them on different subsets of the original dataset, then aggregating their predictions to form a final verdict. It is particularly effective in reducing variance and avoiding overfitting, making it widely used

			in decision tree algorithms and complex classification tasks across various domains.
	RF	Random forests (Breiman, 2001)	An ensemble learning method that builds upon the concept of bagging by creating a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees. It enhances prediction accuracy and controls over-fitting by introducing randomness in the tree generation process through feature and data sampling.
	Boosting	Boosting (Schapire, 1990; Freund and Schapire; 1996)	An ensemble technique that aims to create a strong classifier from a number of weak classifiers. It works by sequentially applying weak models to progressively modified versions of the data, increasing the weight of misclassified instances so that subsequent models focus more on difficult cases. It has widely used variants include AdaBoost and Gradient Boosting, which are effective in reducing bias and variance in complex datasets.
Base reference	Null	Null	A simple model that provides a baseline by using no predictive information to make forecasts in statistics and machine learning. It typically predicts the most frequent category in classification tasks or the mean/median in regression tasks. This model is important for performance benchmarking, as it sets the minimal threshold that any other more complex model should exceed to be considered effective.

2. Methods

The study includes twenty-three algorithms, covering the most widely employed algorithms in the literature. The algorithms can be categorized as instance-based (KNN), statistical (Naive Bayes, Logistic Regression, Partial Least Squares), tree and rule-based (CART, C5.0, C5.0-rules, RuleFit, Bayesian Additive Trees), neural network-based (multilayer perceptron), spline-based (MARS), kernel-based (SVM) and ensemble approaches (Bagging, Boosting, Random Forests). Besides, the Null model is included in the study serving as a benchmark (base) reference as a simple and non-informative model that can be obtained without building any model. It should be noted that different base learner models are utilized in the training process of ensemble models. The C5.0 algorithm, for instance, was not only included in the study on standalone basis but was also considered as a base learner for the bagging algorithm. A similar approach has been carried out for CART, Mars, Mlp algorithms. These algorithms were used as base learner in both bagging and boosting ensemble models. The list of these algorithms and comprehensive explanations are presented in Table 1.

3. Experimental Design and Settings

3.1. Data Description and Source

The dataset was retrieved from a data science platform Kaggle (2023) which is an open source machine learning and data sharing platform. The data set includes thirty variable measurements of one hundred and forty-five students. The sequential grades of the students are considered as the target variable in the study. Variables and their characteristics can be seen in Table 2.

Due to the data set consisting almost completely of categorical data, low-frequency categories were merged to make the results more generalizable and not negatively affect the model estimation. The categories having a frequency of about ten or less were joined with the closest category. Since the target variable is multi-level and the frequency variation between levels is quite volatile (e.g., only seven students failed), we have treated grades below CC, which are defined as failing and conditionally passing, as Fail, and the remaining grades as Success. Therefore, the problem is treated as a binary classification problem. Details of these merging processes are presented in Table 2.

Table 2. Characteristics of the data set

Type	Question	Possible Answers
Personal	Age	(1: 18-21, 2: 22-25, 3: 26+)
	Sex	(1: Female, 2: Male)
	Graduated High School Type	(1: Private, 2: State, 3: Other)
	Scholarship Type	(1: None, 2: 25%, 3: 50%, 4: 75%, 5: Full) **Preprocess**: (None + 25% + 50%) as 50% and lower

	Additional Work	(1: Yes, 2: No)
	Regular Artictic or Sports Activity	(1: Yes, 2: No)
	Do you have a partner?	(1: Yes, 2: No)
	Total salary if available	(1: $135-200, 2: $201-270, 3: $271-340, 4: $341-410, 5: Above $410) **Preprocess:** ($341-410 + Above $410) as $341 and above
	Transportation to the university	(1: Bus, 2: Private Car/Taxi, 3: Bicycle, 4: Other) **Preprocess:** (Bicycle + Other) as Other
	Accommodation type in Cyprus	(1: Rental, 2: Dormitory, 3: With Family)
Family	Mother's education	(1: Primary School, 2: Secondary School, 3: High School, 4: University, 5: Msc., 6: Ph.D.) **Preprocess:** (University + Msc. + Ph.D.) as University
	Father's education	
	Number of sisters/brothers (If available)	(1: 1, 2: 2, 3: 3, 4: 4, 5: 5 or above)
	Parental status	(1: Married, 2: Divorced, 3: Died - One of Them or Both)
	Mother's occupation	(1: Retired, 2: Housewife, 3: Government Officer, 4: Private Sector Employee, 5: Self-Employment, 6: Other) **Preprocess:** (Self-Employment + Other) as Other
	Father's occupation	
Education Habits	Weekly study hours	(1: None, 2: <5 Hours, 3: 6-10 Hours, 4: 11-20 Hours, 5: More Than 20 Hours) **Preprocess:** (11-20 hours + More than 20 hours) as More than 11 hours
	Reading frequency (non-scientific books/journals)	(1: None, 2: Sometimes, 3: Often)
	Reading frequency (Scientific books/journals)	(1: None, 2: Sometimes, 3: Often)
	Attendance to the seminars/conferences related to the department	(1: Yes, 2: No)
	Impact of your projects/activities on your success	(1: Positive, 2: Negative, 3: Neutral)
	Attendance to classes	(1: Always, 2: Sometimes, 3: Never)
	Preparation to midterm exams 1	(1: Alone, 2: With Friends, 3: Not Applicable)
	Preparation to midterm exams 2	(1: Closest Date to The Exam, 2: Regularly During the Semester, 3: Never)
	Taking notes in classes	(1: Never, 2: Sometimes, 3: Always)
	Listening in classes	(1: Never, 2: Sometimes, 3: Always)
	Discussion improves my interest and success in the course	(1: Never, 2: Sometimes, 3: Always)
	Flip-classroom	(1: Not Useful, 2: Useful, 3: Not Applicable)
	Cumulative grade point average in the last semester	(1: <2.00, 2: 2.00-2.49, 3: 2.50-2.99, 4: 3.00-3.49, 5: above 3.49)
	Expected Cumulative grade point average in the graduation	
Output	Grade	(0: Fail, 1: DD, 2: DC, 3: CC, 4: CB, 5: BB, 6: BA, 7: AA) **Preprocess:** (Fail + DD + DC) as Fail; the rest of grades as Success

3.2. Preprocessing and Parameter Tuning

The preprocessing approach and experimental settings applied to the dataset before applying machine learning models can be summarized as follows:

i. The dataset was processed with one-hot encoding and label encoding for nominal and ordinal variables, respectively.

ii. Numerical variables have been standardized.

iii. The dataset is split 75% as training data and 25% as test data. As cross validation approach, the 10-fold CV method was utilized. The models were trained with the data obtained with cross-validation on the training data and their generalization performance (i.e., testing) was evaluated with the test data.

iv. The grid space approach was adopted as the model tuning parametrization. The optimal parameters were derived by using a parameter space consisting of thirty different possible values of the unique parameters of each model. The ranges and optimum values of the tuning parameters for each model are provided in detail in Table 4.

v. The test performance was extracted for each model based on the optimal parameters found by cross-validation.

For the best model among all models in the test performances, confusion matrix, roc curve and variable importance results are presented.

3.3. Performance Criteria

In classification models, depending on whether the target variable is binary or multilevel, performance criteria are primarily defined based on the confusion matrix. A classical confusion matrix can be presented as the following structure given in Table 3.

Table 3. A general representation of a confusion matrix

		Predicted	
		Positive	**Negative**
Actual (Truth)	**Positive**	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

In this study, the accuracy, area under the roc curve (AUC) and F-score, which are the most widely used measures in the literature, can be defined based on confusion matrix as follows.

- **Accuracy**:

  The percentage of correctly classified cases (including true positives and true negatives) relative to the total number of cases is defined as accuracy.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

  The model's predictive skill increases as the accuracy value converges to one.

- **Area Under the Curve (AUC)**:

  In binary classification problems, the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) statistical measure is utilized to assess a model's inherent capacity to differentiate between the positive and negative classes across a range of thresholds for classification. For different threshold values, the true positive rate (sensitivity) is displayed against the false positive rate (1-specificity) through the ROC curve. An AUC value of 1.0 indicates a perfect classifier, while a value of 0.5 indicates a model that performs no better than random chance at classifying true positives and true negatives. The AUC measures the model's overall ability in performing effectively.

- **F-score (or F1-score)**: The F1-score, also known as the F-score or F-measure, is a robust metric for assessing the accuracy of a binary classification model, especially in contexts in which false positives and false negatives have different costs or when class imbalances are present. It is a harmonic mean of precision and recall. The harmonic mean, in contrast to the arithmetic mean, tends to be the lower of the two values, providing that both precision and recall are at an appropriate level. In particular, the F1-score approaches its least accurate value at 0, while reaching its best value at 1, corresponding to perfect precision and recall. The F1-score is defined by using the confusion matrix components as follows:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where

$$\text{Precision} = \frac{TP}{(TP + FP)}$$

$$\text{Recall} = \frac{TP}{(TP + FN)}$$

Table 4. The ranges of parameters corresponding to the each model

Model	**Range**	**Best**
Bag (C5.0)	min_n: [2, 15]	min_n: 6
Bag (CART)	tree_depth: [1, 15], min_n: [2, 15], cost_complexity: [0, 1]	tree_depth: 13, min_n: 6, cost_complexity: 3.2x10^-8
Bag (MARS)	num_terms [0, min(200, max(20, 2 * #variables)) + 1], prod_degree: [1, 2], prune_method: [backward, none, exhaustive, forward, seqrep, cv]	num_terms: 4, prod_degree: 2, prune_method: backward
Bag (MLP)	hidden_units: [2, 20] penalty: [0, 1]	hidden_units: 4 penalty: 0.00000218
BAT	trees: [10, 200] prior_terminal_node_coef: [0.01, 1] prior_terminal_node_expo: [0.01, 2]	trees: 106 prior_terminal_node_coef: 0.0928 prior_terminal_node_expo: 1.70

Boosting (C5.0)	trees: [1, 100] min_n: [2, 15] sample_size: [0.1, 1]	trees: 6 min_n: 7 sample_size: 0.969
Boosting (LightGBM)	mtry: [1, #variables] trees: [1, 2000] min_n: [2, 40] tree_depth: [1, 15] learn_rate: [-3, -0.5] loss_reduction: [-10, 1.5]	mtry: 10 trees: 1080 min_n: 2 tree_depth: 11 learn_rate: 0.00115 loss_reduction: 0.0486
Boosting (XGBoost)	mtry: [1, #variables] trees: [1, 2000] min_n: [2, 40] tree_depth: [1, 15] learn_rate: [-3, -0.5] loss_reduction: [-10, 1.5] sample_size: [0.1, 1]	mtry: 2 trees: 1212 min_n: 2 tree_depth: 5 learn_rate: 0.00990 loss_reduction: 2.62x10^-8 sample_size: 0.706
C5.0	min_n: [2, 15]	min_n: 3
C5.0 Rules	trees: [1, 100] min_n: [2, 15]	trees: 85 min_n: 3
CART	tree_depth: [1, 15], min_n: [2, 15], cost_complexity: [0, 1]	tree_depth: 5, min_n: 7, cost_complexity: 0.0000307
KNN	neighbors: [1, 20] weight_function: [cosine] dist_power: [0.1, 2]	neighbors: 14 weight_function: cosine dist_power: 1.37
Logistic Regression	none	none
MARS	num_terms [0, min(200, max(20, 2 * #variables)) + 1], prod_degree: [1, 2], prune_method: [backward, none, exhaustive, forward, seqrep, cv]	num_terms: 5, prod_degree: 2, prune_method: backward
MLP	hidden_units: [1, 10] penalty: [-10, 0] epochs: [10, 1000]	hidden_units: 2 penalty: 0.00392 epochs: 706
Naive Bayes	smoothness: [0.01, 2], Laplace: [0, 1]	smoothness: 1.33, Laplace: 0.0493
NULL	none	none
PLS	predictor_prop: [0, 1] num_comp: [2, 20]	predictor_prop: 0.0295 num_comp: 4
Random Forests	mtry: [1, 100] trees: [1, 2000] min_n: [2, 40]	mtry: 70 trees: 1648 min_n: 36
RuleFit	mtry: [0, 1] trees: [1, 100] min_n: [1, 20] tree_depth: [1, 20] learn_rate: [0, 1] loss_reduction: [0, 20] sample_size: [0, 2] penalty: [0, 1]	mtry: 0.453 trees: 8 min_n: 6 tree_depth: 6 learn_rate: 5.99*10^-8 loss_reduction: 7.92 sample_size: 0.799 penalty: 0.000883
SVM (Linear)	cost: [0, 30] margin: [0, 1]	cost: 0.244 margin: 0.177
SVM (Polynomial)	cost: [0, 30] degree: [1, 3] scale_factor: [0, 1]	cost: 5.84 degree: 2 scale_factor: 0.000605
SVM (Radial)	cost: [0, 30] rbf_sigma: [0, 1] margin: [0, 1]	cost: 20.4 rbf_sigma: 0.000467 margin: 0.178

4. Results and Discussion

This section presents the performance comparisons of the twenty-three different machine learning methods evaluated in this study. Initially, the performance of these methods on the test data according to measures such as accuracy, AUC and F-score are given in Table 5.

Table 5. The comparative test performance results of machine learning methods

Model	Accuracy	AUC	F-Score
Bag (C5.0)	0.7027	0.7500	0.7027
Bag (CART)	0.6757	0.7794	0.7143
Bag (MARS)	0.6757	0.6824	0.7273
Bagging (MLP)	0.6486	0.7676	0.6977
BAT	0.7027	0.7515	0.7442
Boosting (C5.0)	0.6757	0.7088	0.6842
Boosting (LightGBM)	**0.7568**	**0.7941**	0.7805
Boosting (XGBoost)	0.7027	0.7676	0.7442
C5.0	0.6486	0.6956	0.6667
C5.0 (Rules)	0.7297	0.7676	0.7619
CART	0.7027	0.7324	0.7556
KNN	0.6216	0.7441	0.6667
LR	0.5946	0.5676	0.6512
MARS	0.6486	0.7250	0.6486
MLP	0.6486	0.6941	0.6977
NB	0.5676	0.7500	0.7037
Null	0.5405	0.5000	0.7018
PLS	0.6216	0.7279	0.6818
RF	0.6757	0.7765	0.6842
RuleFit	0.7568	0.7500	**0.7907**
SVM (Linear)	0.5946	0.7176	0.6667
SVM (Polynomial)	0.7027	0.7412	0.7442
SVM (Radial)	0.6486	0.7088	0.6977

According to the results given in Table 5, LightGBM as a boosting algorithm provided the best results in the accuracy (0.7568) and AUC (0.7941) criteria, while RuleFit algorithm dominated in the F1-score (0.7907). It is worth to note that RuleFit algorithm yields slightly higher F1-score than LightGBM algorithm and LightGBM is the second-best algorithm in terms of this criterion. By combining these findings, it can be said that the LightGBM algorithm achieves the most generalizable and superior performance than any other algorithm. A visual interpretation of the AUC values, which are often favored in studies, is also given in Figure 1.

The null model is also included in the study to represent a reference and to clarify the necessity of complex models. In order to assess whether each model is statistically significantly different from each other, especially the null model, DeLong (Delong et al., 1988) and McNemar (McNemar, 1947) tests were performed. The DeLong test relies on AUC values to compare machine learning models, whereas the McNemar test is based on model predictions. The statistical significance value for both tests was set at 0.05 and the results are reported in Table 6.

According to the DeLong test results, all models are statistically different from the Null model, while two bagging models (with CART and MLP learners), LightGBM, XgBoost, RuleFit and SVM (Linear kernel) models have statistically different AUC values with logistic regression. Regarding the McNemar test, LightGBM, as the best model, provided statistically different predictions from MARS, RF and Bagging (C5.0 learner) models, while all model predictions were different from Null and NB models.

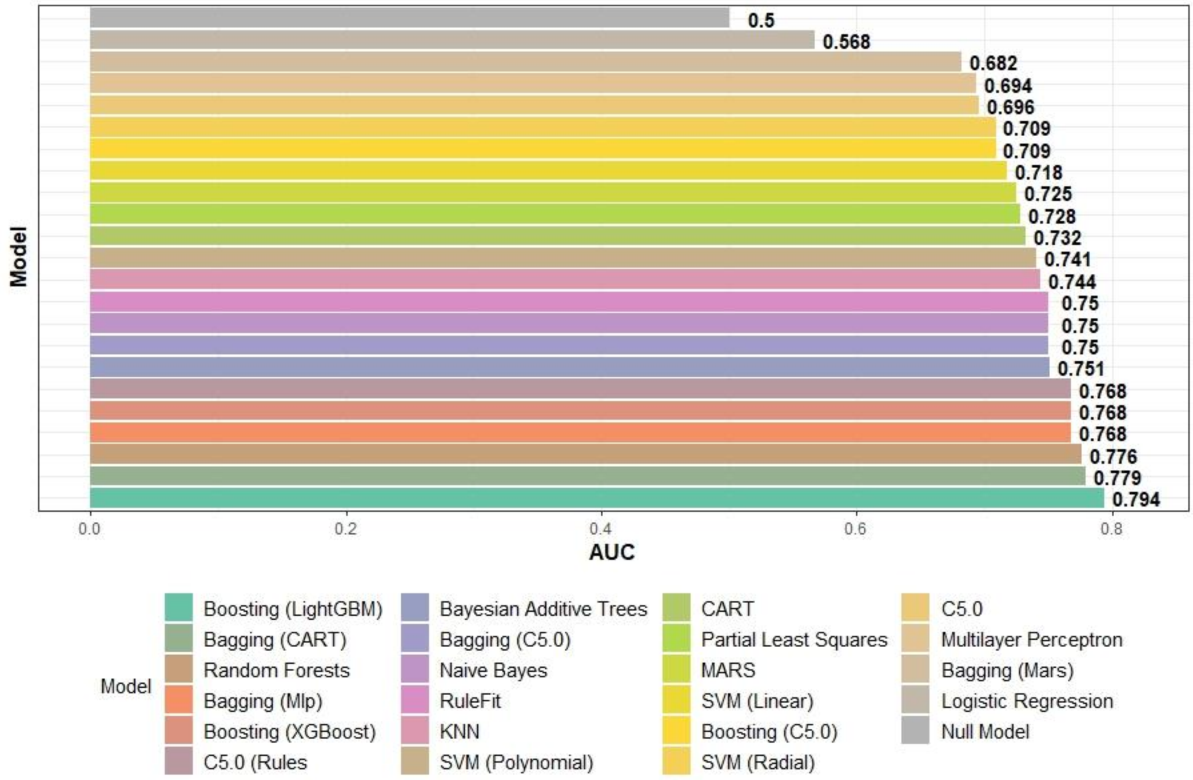


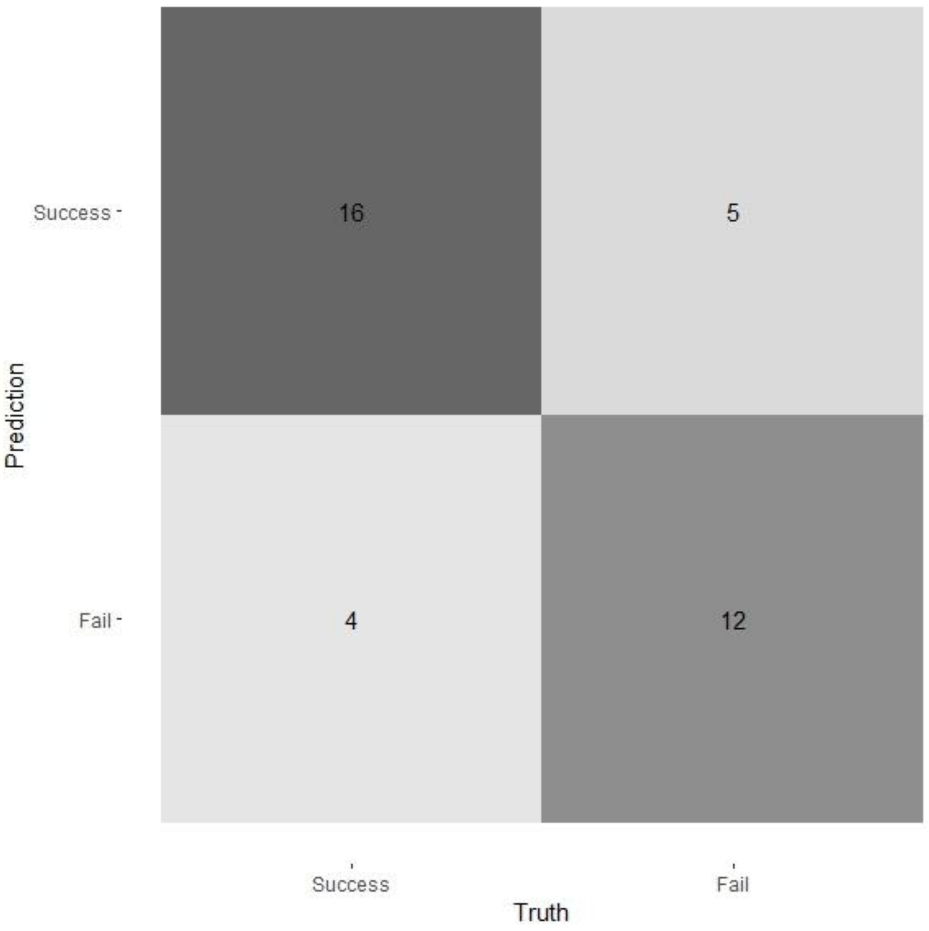
Figure 1. Visual comparison of performance results according to the AUC criterion

Table 6. The statistical comparison of each model based on AUC values and predicted categories

Model	DeLong Test	McNemar Test
Bag (C5.0)	(Null: 0.0019)	(Null: <0.001; NB: 0.0001; SVM (Linear): 0.0433; BAT: 0.0412; Bag (MARS): 0.0233; PLS: 0.0455; Boosting (XGBoost): 0.0133)
Bag (CART)	(Null: 0.0002; LR: 0.038)	(Null: 0.0003; NB: 0.0015)
Bag (Mars)	(Null: 0.0423)	(Null: 0.0009; NB: 0.0094; Bag (C5.0): 0.0233)
Bag (Mlp)	(Null: 0.0012; LR: 0.033)	(Null: 0.0005; NB: 0.0026)
BAT	(Null: 0.0019)	(Null: 0.0005; NB: 0.0026; MARS: 0.0412; Bag (C5.0): 0.0412)
Boosting (C5.0)	(Null: 0.0180)	(Null: <0.001; NB: 0.0002)
Boosting (LightGBM)	(Null: 0.0006; LR: 0.027)	(Null: 0.0015; NB: 0.0077; MARS: 0.0133; RF: 0.0233; Bag (C5.0): 0.0133)
Boosting (XGBoost)	(Null: 0.0001; LR: 0.038)	(Null: 0.0012; NB: 0.0026)
C5.0	(Null: 0.0295)	(Null: 0.0001; NB: 0.0003)
C5.0 (Rules)	(Null: 0.0007)	(Null: 0.0003; NB: 0.0015)
CART	(Null: 0.0101)	(Null: 0.0002; NB: 0.0009)
KNN	(Null: 0.0026)	(Null: 0.0003; NB: 0.0033)
LR	(Null: 0.4552)	(Null: 0.0005; NB: 0.0098)
MARS	(Null: 0.0070)	BAT: 0.0412, PLS: 0.0455; Boosting (LightGBM): 0.0133)
MLP	(Null: 0.0410)	(Null: 0.0005; NB: 0.0026)
NB	(Null: 0.0024)	(Null: 0.2482; LR: 0.0098)
Null	None	None
PLS	(Null: 0.0076)	(Null: 0.0009; NB: 0.0044; MARS: 0.0455; RF: 0.0412; Bag (C5.0): 0.0455)

RF	(Null: 0.0004)	(Null: <0.001; NB: 0.0002; PLS: 0.0412; Boosting (LightGBM): 0.0233)
RuleFit	(Null: 0.0050; LR: 0.042)	(Null: 0.0005; NB: 0.0055)
SVM (Linear)	(Null: 0.0156; LR: 0.048)	(Null: 0.0015; NB: 0.0159; Bagging (C5.0): 0.0433)
SVM (Polynomial)	(Null: 0.0047)	(Null: 0.0005; NB: 0.0026)
SVM (Radial)	(Null: 0.0228)	(Null: 0.0005; NB: 0.0026)

It is important that the models are statistically different from each other, especially from the Null model, for the generalizability and usability of the results. In this context, we focus on the predictions of the LightGBM model, which is found to be the best model, and the confusion matrix and ROC cuver derived from these predictions is given in Figure 2 and 3, respectively.



Prediction \ Truth	Success	Fail
Success	16	5
Fail	4	12

Figure 2. Confusion matrix for the best model (Boosting (LightGBM))

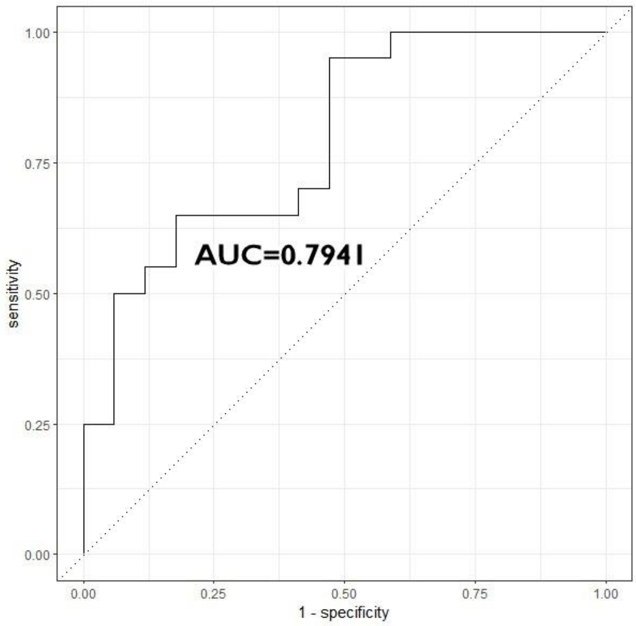


Figure 3. Roc curve for the best model (Boosting (LightGBM))

The confusion matrix and ROC curve suggest that the LightGBM model provides promising results in predicting student performance. It is critical to identify the most important factors for the performance of the model, i.e., for discriminating between success and failure. Therefore, the variable importance plot computed by using the intrinsic variable importance scores of the LightGBM model is displayed in Figure 4.

In Figure 4, the fifteen most important variables are ranked on a scale of 0-100. According to this graph, variables such as last semester GPA between 2-2.50, average income between $135-200, expected GPA between 3-3.50, gender of the student being male, number of brothers or sisters appear to be the most important variables in affecting success. Likewise, variable levels such as attending seminars etc. and not having a partner were also found to be important in model performance. The results and particularly variable importance scores presented in this study may provide a more valuable set of sociological and academic insights for researchers studying in the field of education.

It can be said that the findings of the study provide better performance by using a broader method compared to the studies in the literature such as Yılmaz and Sekeroglu (2020), Chen and Zhai (2023). In Asselman et al. (2023), the XGBoost algorithm, one of the ensemble approaches, stands out as an ensemble method and demonstrates similar performance to our study, but its shortcomings are notable in terms of variable importance and statistical significance tests. In Adejo and Connoly's (2018) study, the hybrid machine learning model also produced a competitive result and concluded that university support had a significant impact on success. In this context, it can be said that it is compatible with the scholarship status in our study.

The finding that students' performance in previous semesters has a significant effect on their future achievement is consistent with the literature (Pallathadka et al., 2021). Similarly, the findings that income and family education have a significant effect on achievement supports the results of Filho et al. (2023).

The gender variable, which was found to be relatively significant in the study, stands out as a different finding from the study of Karaboğa and Demir (2023). On the other hand, Suleiman and Anane (2022) reported that gender was a significant but low contributing variable on student achievement. As in our study, last semester GPA was considered significant in this study as well.

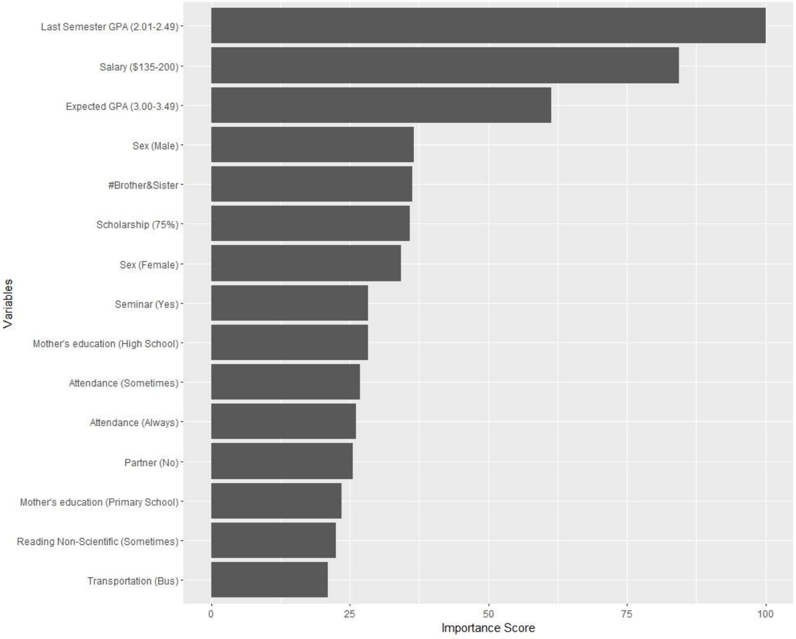


Figure 4. Variable importance plot based on Boosting (LightGBM) model.

5. Conclusion

In this study, a comprehensive comparison of the performance of machine learning methods is presented both in terms of classical metrics and statistically. Machine learning algorithms, which are widely used in the field of education as in every field, have been shared to determine the extent to which they differ statistically and the ways to determine the importance of variables rather than their performance alone. The LightGBM algorithm was ranked as the best algorithm by cross-validating twenty-three algorithms using a real dataset based on comparison them on accuracy, AUC, and F-score criteria. The results were statistically compared with two different tests to investigate the extent to which the best method differs, and it was found that LightGBM provided favorable results in this respect as well. In addition, the confusion matrix, ROC curve and variable importance plots indicated that the LightGBM algorithm offers generalizable performancealong with identifying the relative importance of the most important factors affecting student achievement.

The study is not free of limitations. First, the implementation of deep learning algorithms in such studies may provide useful insights. Furthermore, a field-based analysis of student achievement performances and the factors affecting them may provide more effective results in different subgroups. In future work, we would like to address these two limitations, and we aim to specialize the most advanced deep learning models to narrower focused educational groups.

References

Adejo, O. W., & Connolly, T. (2018). Predicting student academic performance using multi-model heterogeneous ensemble approach. Journal of Applied Research in Higher Education, 10(1), 61–75. https://doi.org/10.1108/JARHE-09-2017-0113

Alalawi, K., Athauda, R., & Chiong, R. (2023). Contextualizing the current state of research on the use of machine learning for student performance prediction: A systematic literature review. Engineering Reports, 5(12), e12699. https://doi.org/10.1002/eng2.1269

Albreiki, B., Zaki, N., & Alashwal, H. (2021). A Systematic Literature Review of Student' Performance Prediction Using Machine Learning Techniques. Education Sciences, 11(9), Article 9. https://doi.org/10.3390/educsci11090552

Asselman, A., Khaldi, M., & Aammou, S. (2023). Enhancing the prediction of student performance based on the machine learning XGBoost algorithm. Interactive Learning Environments, 31(6), 3360–3379. https://doi.org/10.1080/10494820.2021.1928235

Breiman, L. (1996). Bagging predictors. Machine learning, 24, 123-140.

Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. Classification and Regression Trees (CART). 1984. Belmont, CA, USA: Wadsworth International Group.

Chen, Y., & Zhai, L. (2023). A comparative study on student performance prediction using machine learning. Education and Information Technologies, 28(9), 12039–12057. https://doi.org/10.1007/s10639-023-11672-1

Chipman, H. A., George, E. I., & McCulloch, R. E. (2010). BART: Bayesian additive regression trees.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), 21-27.

Cox, D. R. (1958). The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological), 20(2), 215–232.

DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. Biometrics, 837-845.

Deo, R. C., Yaseen, Z. M., Al-Ansari, N., Nguyen-Huy, T., Langlands, T. A. M., & Galligan, L. (2020). Modern Artificial Intelligence Model Development for Undergraduate Student Performance Prediction: An Investigation on Engineering Mathematics Courses. IEEE Access, 8, 136697–136724. https://doi.org/10.1109/ACCESS.2020.3010938

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine learning, 29, 103-130.

Elbadrawy, A., Polyzou, A., Ren, Z., Sweeney, M., Karypis, G., & Rangwala, H. (2016). Predicting Student Performance Using Personalized Analytics. Computer, 49(4), 61–69. https://doi.org/10.1109/MC.2016.119

Filho S., , R. L. C., Brito, K., & Adeodato, P. J. L. (2023). A data mining framework for reporting trends in the predictive contribution of factors related to educational achievement. Expert Systems with Applications, 221, 119729.

Freund, Y., & Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. In icml (Vol. 96, pp. 148-156).

Friedman, J. H. (1991). Multivariate adaptive regression splines. The annals of statistics, 19(1), 1-67.

Friedman, J. H., & Popescu, B. E. (2008). Predictive learning via rule ensembles.

Gamulin, J., Gamulin, O., & Kermek, D. (2016). Using Fourier coefficients in time series analysis for student performance prediction in blended learning environments. Expert Systems, 33(2), 189–200. https://doi.org/10.1111/exsy.12142

Guan, C., Mou, J., & Jiang, Z. (2020). Artificial intelligence innovation in education: A twenty-year data-driven historical analysis. International Journal of Innovation Studies, 4(4), 134–147. https://doi.org/10.1016/j.ijis.2020.09.001

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural networks, 2(5), 359-366.

Hussain, M., Zhu, W., Zhang, W., Abidi, S. M. R., & Ali, S. (2019). Using machine learning to predict student difficulties from learning session data. Artificial Intelligence Review, 52(1), 381–407. https://doi.org/10.1007/s10462-018-9620-8

Karaboğa, H. A., & Demir, I. (2023). Examining the factors affecting students' science success with Bayesian networks. International Journal of Assessment Tools in Education, 10(3), 413-433.

Liu, J., Loh, L., Ng, E., Chen, Y., Wood, K. L., & Lim, K. H. (2020). Self-Evolving Adaptive Learning for Personalized Education. Conference Companion Publication of the 2020 on Computer Supported Cooperative Work and Social Computing, 317–321. https://doi.org/10.1145/3406865.3418326

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika, 12(2), 153-157.

Pallathadka, H., Wenda, A., Ramirez-Asís, E., Asís-López, M., Flores-Albornoz, J., & Phasinam, K. (2023). Classification and prediction of student performance data using various machine learning algorithms. Materials Today: Proceedings, 80, 3782–3785. https://doi.org/10.1016/j.matpr.2021.07.382

Quinlan, J. R. (1992). Learning with continuous classes. In 5th Australian joint conference on artificial intelligence (Vol. 92, pp. 343-348).

Quinlan, J. R. (1993). Combining instance-based and model-based learning. In Proceedings of the tenth international conference on machine learning (pp. 236-243).

Sağlam, A. Ç., & Aydoğmuş, M. (2016). Gelişmiş ve Gelişmekte Olan Ülkelerin Eğitim Sistemlerinin Denetim Yapıları Karşılaştırıldığında Türkiye Eğitim Sisteminin Denetimi Ne Durumdadır? Uşak Üniversitesi Sosyal Bilimler Dergisi, 9(1), 17–38. https://dergipark.org.tr/en/pub/usaksosbil/issue/21662/232993

Schapire, R. E. (1990). The strength of weak learnability. Machine learning, 5, 197-227.

Schölkopf, B., & Smola, A. J. (2002). Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press.

Sekeroglu, B., Abiyev, R., Ilhan, A., Arslan, M., & Idoko, J. B. (2021). Systematic Literature Review on Machine Learning and Student Performance Prediction: Critical Gaps and Possible Remedies. Applied Sciences, 11(22), Article 22. https://doi.org/10.3390/app112210907

Students Performance. (2023) Retrieved 25 September 2023, from https://www.kaggle.com/datasets/joebeachcapital/students-performance

Suleiman, R., & Anane, R. (2022). Institutional Data Analysis and Machine Learning Prediction of Student Performance. 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 1480–1485. https://doi.org/10.1109/CSCWD54268.2022.9776102

Tilahun, L. A., & Sekeroglu, B. (2020). An intelligent and personalized course advising model for higher educational institutes. SN Applied Sciences, 2(10), 1635. https://doi.org/10.1007/s42452-020-03440-4

Tran, T.-O., Dang, H.-T., Dinh, V.-T., Truong, T.-M.-N., Vuong, T.-P.-T., & Phan, X.-H. (2017). Performance Prediction for Students: A Multi-Strategy Approach. Cybernetics and Information Technologies, 17(2), 164–182. https://doi.org/10.1515/cait-2017-0024

Vapnik, V., Golowich, S., & Smola, A. (1996). Support vector method for function approximation, regression estimation and signal processing. Advances in neural information processing systems, 9.

Wold, H. (1982). Soft modelling: the basic design and some extensions. Systems under indirect observation, Part II, 36-37.

Wold, S., Ruhe, A., Wold, H., & Dunn, Iii, W. J. (1984). The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses. SIAM Journal on Scientific and Statistical Computing, 5(3), 735-743.

Wu, Z., He, T., Mao, C., & Huang, C. (2020). Exam paper generation based on performance prediction of student group. Information Sciences, 532, 72–90. https://doi.org/10.1016/j.ins.2020.04.043

Yousafzai, B. K., Hayat, M., & Afzal, S. (2020). Application of machine learning and data mining in predicting the performance of intermediate and secondary education level student. Education and Information Technologies, 25(6), 4677–4697. https://doi.org/10.1007/s10639-020-10189-1

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

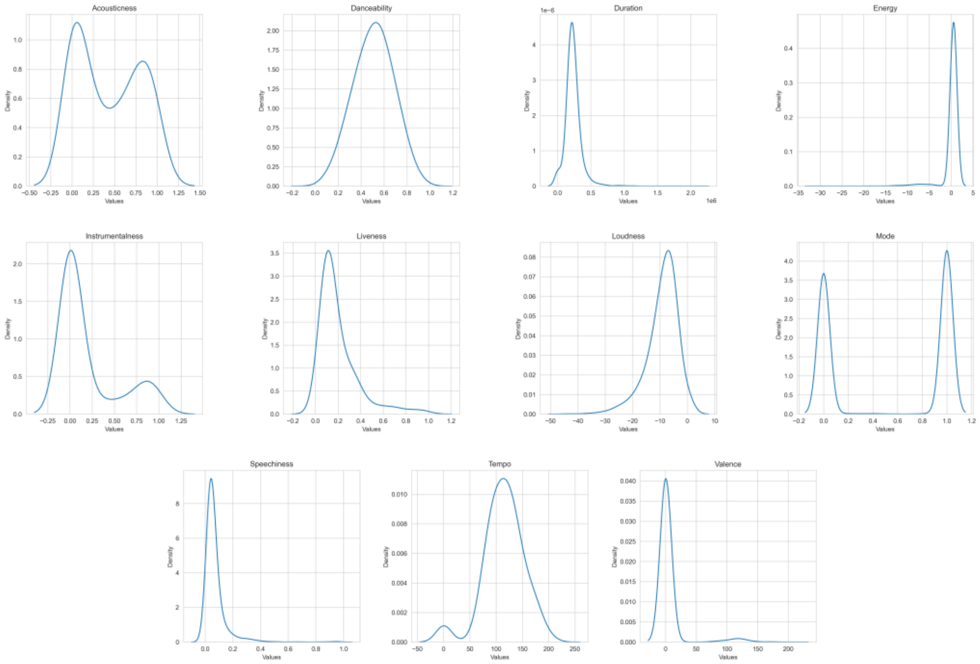


Figure 1. Distribution of the Attributes

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.

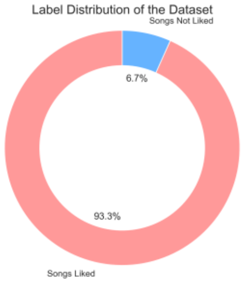


Figure 2. Label Distribution of the Dataset

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.

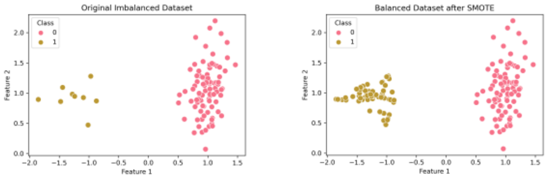
2.2. SMOTE

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, λ 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.

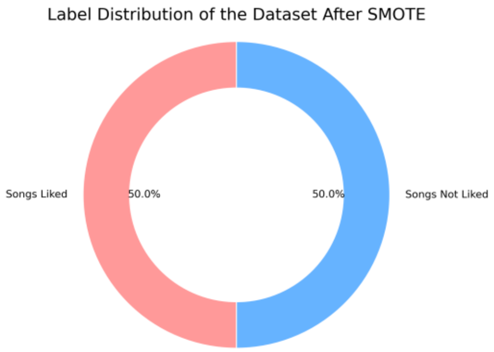


Figure 4. Label Distribution of the Dataset after SMOTE

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 γ 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%

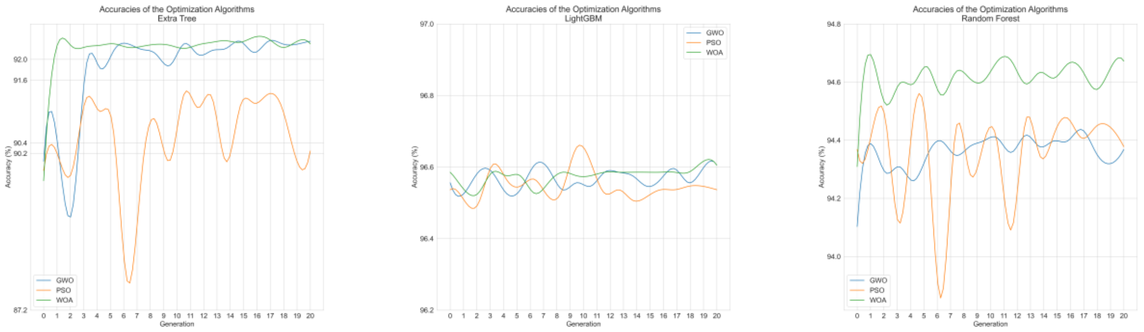
Table 5. Optimized Parameters

	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.

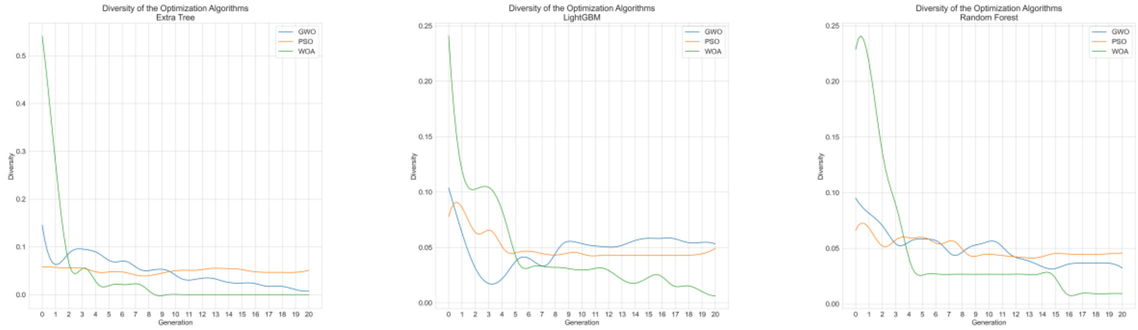
To analyze the performance of the optimizers in depth, a thorough analysis was conducted. Figure 5 shows the highest accuracy attained by each model across all generations.



(a) (b) (c)

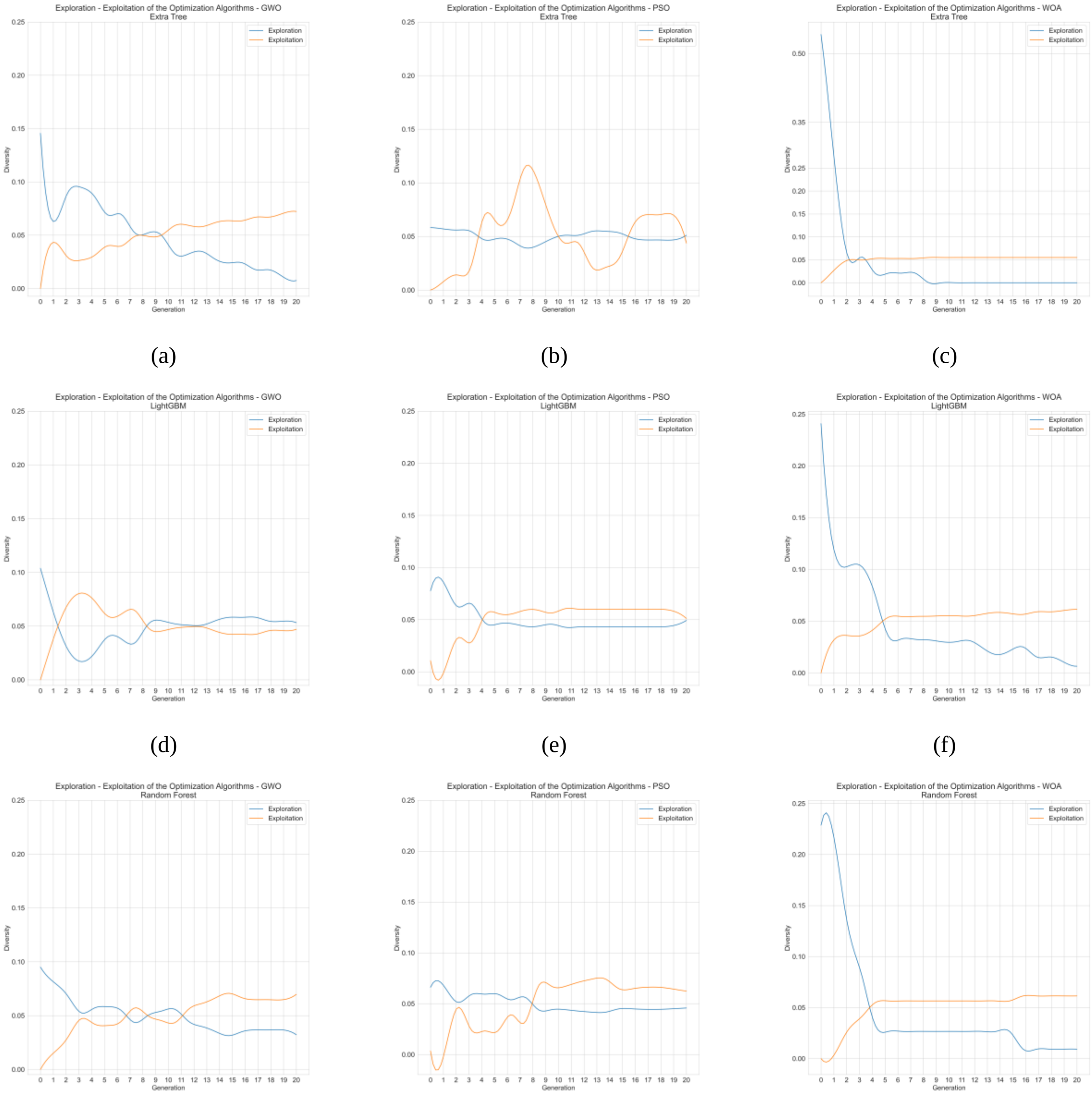
Figure 5. Accuracy Graph of the Optimization Algorithms. (a) Extra Tree (b) LightGBM (c) Random Forest

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 5th 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 5th generation, whereas divergence seemed to be much less for PSO after 13th 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.

Araştırma Makalesi
Zeki Sistemler Teori ve Uygulamaları Dergisi 7(2) (2024) 66-80
DOI: 10.38016/jista.1427938

Tedarik Zincirinde Hibrit Talep Tahmin Modeli Önerisi: Çelik Sektörü Uygulaması

Orhan Torkul[1], Erhan Kor[2], Merve Şişci[3*]

[1, 2, 3] Endüstri Mühendisliği Bölümü, Sakarya Üniversitesi, Sakarya, Türkiye

torkul@sakarya.edu.tr, erhneng@outlook.com, mervesisci@sakarya.edu.tr

Öz

Uzun imalat süreleri, süreç içi stokların yüksek olması ve tezgahlardan yararlanma oranlarının düşük olması üretim sistemlerinde karşılaşılan önemli planlama problemlerindendir. Bunların içerisinde, imalat sürelerinin uzun olması dolayısıyla sipariş gecikmelerinin meydana gelmesi önemli problem alanlarından birisidir. Bu çalışmada, çelik sektöründe sipariş gecikmelerinin sebepleri araştırılarak bunların ortadan kaldırılması ile tedarik zincirinde sürekliliğin sağlanması için bir talep tahmini modeli önerisi geliştirilmesi amaçlanmıştır. Önerilen model, ürünler için ihtiyaç duyulan ve sipariş gecikmelerinde birincil derecede önemli olan hammadde ve yarı mamulün ihtiyaç duyulan zamanda ve miktarda belirlenebilmesi için nitelik seçimi ve makine öğrenmesi algoritmalarına dayalı hibrit bir yapıdadır. Geçmiş dönem satış miktarlarının yanı sıra enerji maliyetleri, çelik hammadde fiyatı ve Euro/Dolar paritesi modele bağımsız değişkenler olarak dahil edilmiştir. Talep tahmin modellerinin geliştirilmesinde en ilgili özelliklerin belirlenebilmesi amacıyla 6 farklı nitelik seçimi yöntemi uygulanmıştır. Modeller 3 farklı makine öğrenmesi algoritması ile eğitilmiştir. Geliştirilen modeller çelik sektöründe faaliyet gösteren bir firmanın 4 ürününün 89 aylık verileri üzerinde uygulanmıştır. Deneysel sonuçlara göre, nitelik seçimi yöntemlerinin genel olarak tahmin modellerinin performansını arttırdığı sonucuna ulaşılmasına rağmen, her bir ürün için en uygun tahmin performansını gösteren nitelik kümesi ve talep tahmini yöntemi kombinasyonunun farklılık gösterdiği değerlendirilmiştir. Geliştirilen modeller sayesinde ürünler için sırasıyla %93.6, %94.7, %90.3 ve %91.5 tahmin doğruluğu değerine ulaşılmıştır.

Anahtar Kelimeler: Talep Tahmini, Makine Öğrenmesi, Nitelik Seçimi, Optimizasyon Algoritmaları.

A Proposal of Hybrid Demand Forecasting Model in Supply Chain: Steel Industry Application

Abstract

Long manufacturing times, high in-process stocks and low machine utilization rates are important planning problems encountered in production systems. Among these, order delays due to long manufacturing times are one of the important problem areas. In this study, it is aimed to investigate the reasons for order delays in the steel industry and to develop a demand forecasting model proposal to eliminate them and ensure continuity in the supply chain. The proposed model has a hybrid structure based on feature selection and machine learning algorithms in order to determine the raw materials and semi-finished products needed for products and which are of primary importance in order delays, at the required time and quantity. In addition to past sales amounts, energy costs, steel raw material price and Euro/Dollar parity were included in the model as independent variables. In order to determine the most relevant features in development of demand forecasting models, 6 different feature selection methods were applied. The models were trained with 3 different machine learning algorithms. The developed models were applied on 89-month data of 4 products of a company operating in the steel industry. According to the experimental results, although it was concluded that feature selection methods generally increased performance of forecasting models, it was evaluated that combination of feature set and demand forecasting method showing the most appropriate forecasting performance for each product differed. By the agency of the developed models, 93.6%, 94.7%, 90.3% and 91.5% prediction accuracy values were achieved for products, respectively.

Keywords: Demand Forecasting, Machine Learning, Feature Selection, Optimization Algorithms.

[*] Sorumlu yazar.
E-posta adresi: mervesisci@sakarya.edu.tr

Alındı	: 29 Ocak 2024
Revizyon	: 11 Mart 2024
Kabul	: 18 Nisan 2024

1. Giriş (Introduction)

Günümüzde meydana gelen hızlı değişimler nedeniyle, işletmelerin gelecekteki süreçler için tahmin çalışmaları gerçekleştirmeleri varlıklarını devam ettirebilmeleri için temel bir ihtiyaç haline gelmiştir (Güven, 2020). Müşteri hacimlerindeki büyümeler ve teknolojideki gelişmeler gibi sebeplerle pazar talebine hızlı cevap verebilmek ve müşteriler için sürdürülebilir bir tedarik zinciri sağlayabilmek hayati bir önem taşımaktadır (Keung vd., 2021). Etkili bir tedarik zinciri yönetimi yüksek rekabetli bir ortamda bile işletmenin sürekliliğini güvence altına alabilmektedir. Ancak, özellikle çelik üretimi yapan firmalar, pazar ve talep dalgalanmaları ile karşı karşıya kalabilmektedir (Lee vd., 1997). Bununla birlikte, değişken müşteri talebi tedarik zinciri yönetimini istikrarsızlaştırmakta ve envanter yönetiminde zorluklar oluşturmaktadır. Ürünlerin müşteriye teslimatının zamanında yapılamaması sadece stok seviyelerini etkileyip ek maliyetlere sebep olmakla kalmamakta, aynı zamanda müşterinin beklentisinin ve ürün satın alma motivasyonunun düşmesine de neden olabilmektedir. Bu nedenlerden dolayı, özellikle Endüstri 4.0 çağında teslimat gecikmeleri, ele alınması ve çözülmesi gereken önemli sorunlardan birisidir (Keung vd., 2021). Bahsedilen ihtiyaçlar doğrultusunda, talep tahmini ve satış tahmini üreticilerin, dağıtıcıların ve ticaret yapan firmaların en önemli fonksiyonlarından biri haline gelmiştir (Kochak ve Sharma, 2015).

Tedarik zincirindeki esnekliğini ve dayanıklılığını arttırmak, şirketlerin önemli hedefleri arasında yer almaktadır (El Filali vd., 2022). Bu doğrultuda, talep tahmin yöntemleri üretim planlama, maliyet, stok ve teslimat süreleri gibi birçok konuda önemli avantajlar sunmaktadır. Talep ve tedarik dengesi sağlanarak envanter fazlalığının veya yetersizliğinin azaltılması ile firma karlılığının artışı sağlanabilmektedir (Kochak ve Sharma, 2015). Bu nedenle, literatürde minimum tahmin hatasına sahip talep tahmin modellerinin geliştirilmesi üzerinde yoğun bir şekilde çalışılmaktadır. Geleneksel tahmin yöntemlerinde çok sayıda insan faktörü ve yüksek miktarda hata bulunmaktadır. Aynı zamanda hata toleransları da düşüktür. Bu da doğru tahmin sonuçları almayı zorlaştırmaktadır (Xu ve Wang, 2022). Bazı çalışmalarda (Feizabadi, 2022), geleneksel tahmin yöntemleri ile karşılaştırıldığında makine öğrenmesi modellerinin daha iyi tahmin performansı gösterdiği kanıtlanmıştır.

Makine öğrenmesi modellerinin geliştirilmesinde nitelik seçimi, ilgili özellikleri seçmek ve gürültülü ve alakasız olanları kaldırmak için kritik bir süreç olarak kabul edilmektedir (Elgamal vd., 2020). Talep tahmini konusunun bir alt kümesi olduğu regresyon problemlerinde minimum sayıda özelliğin seçimi hesaplama karmaşıklığını azaltırken, optimum özeliğin seçimi regresyon modellerinin doğruluğunu korumaya yardımcı olur (Ismael vd., 2021). Özellikle en iyi çözümleri sağlamak için popülasyonun değerli bilgisinden yararlanma özelliğine sahip olan evrimsel algoritmalar, özellik optimizasyonu problemleriyle başa çıkmadaki performanslarından dolayı son yirmi yılda oldukça başarı elde etmektedir (Thawkar, 2022). Bu çalışmada, çelik imalat sektörüne ait ürünler için nitelik seçimi ve makine öğrenmesi algoritmalarına dayanan talep tahmin modellerinin geliştirilmesi amaçlanmıştır. Aynı zamanda nitelik seçimi için uyarlanan ve doğadan ilham alan 6 sarmalayıcı optimizasyon algoritmasının talep tahmin modellerinin performansları üzerindeki etkisinin incelenmesi hedeflenmiştir. Bu hedefler doğrultusunda, çelik sektöründe faaliyet gösteren bir firmadan alınan ve çeşitli nitelik oluşturma işlemleri ile elde edilen veri setleri üzerinde, Parçacık Sürüsü Optimizasyonu Algoritması, Harris Şahinleri Optimizasyonu Algoritması, Gri Kurt Optimizasyonu Algoritması, Yusufçuk Optimizasyonu Algoritması, Genetik Optimizasyonu Algoritması ve Yerçekimi Arama Optimizasyonu Algoritması olmak üzere 6 nitelik seçimi yöntemi ile tahmin modellerinde etkili nitelikler belirlenmiştir. Belirlenen 4 ürün için tüm nitelikler ve nitelik seçimi yöntemleri ile belirlenen nitelikler kullanılarak lineer regresyon, yapay sinir ağları ve karar ormanı algoritmaları ile 84 farklı talep tahmini modeli geliştirilmiş ve modellerin performansları karşılaştırılmıştır.

Çalışmanın geri kalanı şu şekilde düzenlenmiştir: Bölüm 2'de literatürde bulunan talep tahmini çalışmaları incelenmektedir. Bölüm 3'te, talep tahmini modellerinin geliştirilmesi için çalışmada kullanılacak olan yöntemler sunulmaktadır. Bölüm 4'te çalışmada önerilen talep tahmini metodolosijinin çelik imalatı sektörüne ait ürünlerden elde edilen veriler üzerinde uygulaması gerçekleştirilmektedir. Bölüm 5'te ise sonuçlar ve tartışma yer almaktadır.

2. Literatür (Literature)

Üretim planlamasına temel oluşturan talep tahmini (Kück ve Freitag, 2021), tedarik zincirindeki belirsizliklerin azaltılması ve böylece tedarik zinciri performansının iyileştirilmesinde kritik bir rol üstlenmektedir. Dolayısıyla talep tahmini enerji, sağlık, otomotiv, tekstil, e-ticaret ve perakende dahil olmak üzere birçok alanda üzerinde yoğun bir şekilde çalışılan bir konudur. Bu bölümde talep tahmini üzerine yapılan çalışmalar incelenmektedir.

Merkuryeva vd. (2019), toptancıdan dağıtıcıya ilaç ürünlerinde örnek olay üzerinden talep tahmin çalışması gerçekleştirmişlerdir. Modeller basit hareketli ortalama, lineer regresyon ve sembolik regresyon yöntemleriyle geliştirilmiştir. Modellerin karşılaştırılması sonucu sembolik regresyon yönteminin en uygun yöntem olduğu sonucuna varılmıştır. Türk ve Kiani (2019), yapmış oldukları çalışmada, Türkiye'deki toplam beyaz eşya satışlarını tahmin etmek için 2007-2015 dönemine ait beyaz eşya satış verilerini kullanarak yapay sinir ağları ve regresyon modelleri geliştirmişlerdir. Fanoodi vd., (2019) çalışmasında, sağlık sistemindeki tedarik zincirinde kan trombosit taleplerinin tahmin edilmesi amaçlanmıştır. Sekiz farklı kan trombositi için 2013-2018 yılları arasındaki günlük talep verileri üzerinde yapay sinir ağları ve Otoregresif Entegre Hareketli Ortalama (ARIMA) ile tahmin modelleri oluşturulmuştur. Aydın ve Yazıcıoğlu (2019), bir süpermarketin kasap reyonunda 3 farklı et türü için Ocak 2017- Aralık 2018 dönemindeki haftalık satış verileri ile talep tahmin modelleri oluşturmuşlardır. Modellerin oluşturulmasında ARIMA ve yapay sinir ağları yöntemlerini kullanmışlardır. Modellerin tahmin performanslarının karşılaştırılması sonucunda yapay sinir ağları modelinin daha iyi tahmin gücüne sahip olduğu gözlemlenmiştir. Torun ve Deste (2021) çalışmasında, Samsun Devlet Hastanesi Ortopedi Bölümü'nde en fazla ihtiyaç duyulan 9 farklı sağlık malzemesine ait 2015-2018 yılları arasındaki verileri kullanılarak talep tahmini çalışması gerçekleştirilmiştir. Modellerin geliştirilmesinde kullanılan yöntemler Hareketli Ortalama, Üstel Düzeltme, Holt'un Doğrusal Yöntemi, Çarpımsal Holt-Winters, Toplamsal Holt-Winters ve Basit Doğrusal Regresyon yöntemleri olmuştur. Tavukçu ve Sennaroğlu (2021) çalışmasında, iş makineleri için yedek parça satışı yapan bir firmadaki stok siparişlerinin tahmin değerlerini elde etmek için 36 aylık talep verileri kullanılmıştır. Hareketli Ortalama, Holt-Winters Metodu ve ARIMA yöntemleri ile tahminler gerçekleştirilmiştir. Mohan vd. (2021), tedarik zincirinde talep tahmini ve rota optimizasyonunu amaçlamışlardır. Depo ürünlerine yönelik talep tahmini modeli için ARIMA yöntemini kullanmışlardır. Kück ve Freitag (2021) çalışmasında, bir imalat şirketinin aylık müşteri taleplerinin tahmini amacıyla K-En Yakın Komşu (KNN) yöntemi kullanılmıştır. Yerel ortalama sabiti, yerel medyan sabiti, dört farklı düzenleme metodu ve çeşitli parametre kombinasyonlarıyla yüksek doğruluk elde edilmesi hedeflenmiştir. Han vd. (2022) çalışmasında, Holt Winters ve Yapay Sinir Ağları yöntemleri kullanılarak Türkiye'deki sıfır otomobil satış değerlerinin tahmin değerleri elde edilmiştir. 2015-2020 arasındaki aylık veriler ve döviz kuru, tüketici güven endeksi, gayrisafi yurt içi hasıla (GSYHİ), reel kesim güven endeksi bağımsız değişkenleri kullanılmıştır. İmece ve Beyca (2022) çalışmasında ilaç firmasına ait bir ürün için Holt Winters, Ridge Regresyon, Rastgele Orman ve Aşırı Gradyan Artırma (XGBoost) yöntemlerini ve bu yöntemlerin kombinasyonlarını kullanarak talep tahmini modelleri oluşturulmuştur. Modellerin geliştirilmesinde ve değerlendirilmesinde 2016-2018 yılları arası günlük ürün satış değerleri kullanmıştır. Feizabadi (2022), yapmış olduğu çalışmada tedarik zinciri performansını iyileştirmek amacıyla Yapay Sinir Ağları ve Dışsal Değişkenli Otoregresif Entegre Hareketli Ortalama (ARIMAX) yöntemlerine dayanan hibrit talep tahmini modelleri geliştirmiştir. Çalışmada çelik imalat firmasından elde edilen veriler kullanılmıştır. Yapılan değerlendirmeler, geleneksel yöntemlere göre yapay zeka tabanlı yöntemlerin daha iyi tahmin sonuçları verdiğini göstermiştir. El Filali vd. (2022), çalışmalarında elektrik ürünleri üzerine talep tahmini için Tekrarlayan Sinir Ağları, Uzun Kısa Süreli Bellek ve Geçitli Tekrarlayan Birim olmak üzere Yapay Sinir Ağları yöntemlerini karşılaştırmışlardır. Geçitli Tekrarlayan Birim modeli en iyi tahmin sonuçlarını sağlayan model olmuştur. Acı ve Doğansoy (2022) çalışmasında Türkiye'de bulunan bir süpermarketin 2 yıllık e-ticaret verileri kullanılarak ürün satış tahmini gerçekleştirilmiştir. En iyi tahmin performansı gösteren modelin elde edilebilmesi için Yapay Sinir Ağları, Derin Öğrenme, Regresyon Ağacı, Gauss Süreç Regresyonu, Topluluk Öğrenme ve Destek Vektör Regresyonu yapay zeka algoritmaları ile modeller oluşturulmuştur. Orzechowski vd. (2023), elektrikli araçların halka açık şarj istasyonlarındaki şarj taleplerinin bir hafta öncesinden tahmin edilmesini amaçlamışlardır. Çalışmada Yapay Sinir Ağları, ARIMA, Rastgele Orman, Destek Vektör Regresyonu ve K-En Yakın Komşu algoritması ile farklı modeller oluşturulmuştur. Modellerin performansları karşılaştırıldığında Yapay Sinir Ağlarının en iyi performansı sağladığı görülmüştür. Yani ve Aamer (2023), yapmış oldukları çalışmada ilaç tedarik zincirinde talep tahmini üzerinde çalışmışlardır. Çalışmada, çok uluslu ilaç şirketlerinden alınan dokuz farklı ürünün verileri kullanılmıştır. Tahmin modelleri için Gradyan Arrtırma Ağaçları, Rasgele Orman, Lineer Regresyon, Polinomiyal Regresyon, Basit Ağaç ve Ağaç Topluluğu makine öğrenmesi yöntemleri kullanılmıştır.

Bu çalışmada, talep tahmini literatüründe az karşılaşılan çelik sektörüne ait ürünler üzerinde talep tahmini çalışması gerçekleştirilmiştir. Talep tahmini modelleri için Lineer Regresyon, Yapay Sinir Ağları ve Karar Ormanı makine öğrenmesi algoritmaları kullanılmıştır. Modellerin performanslarının arttırılması amacıyla optimizasyona dayalı 6 farklı nitelik seçimi yönteminden yararlanılmıştır. Geliştirilen nitelik seçimi-makine öğrenmesi hibrit modellerinin etkinlikleri 4 farklı ürün üzerinde değerlendirilmiştir.

3. Metot (Method)

Bu bölümde çalışmada yararlanılan makine öğrenmesi algoritmaları ve nitelik seçimi yöntemleri açıklamalarına yer verilmiştir.

3.1. Makine Öğrenmesi Regresyon Algoritmaları (Machine Learning Regression Algorithms)

3.1.1. Lineer regresyon (Linear regression)

Bağımsız değişken ile bağımlı değişken arasındaki etkileşim ve bağlantının araştırıldığı yöntem regresyon analizidir. Tahmin modelinde bağımlı değişkene etki eden bir bağımsız değişken varsa bu regresyon tek değişkenli regresyon analizi adlandırılmaktadır. Bağımlı değişkene etki eden birden fazla bağımsız değişkenin olduğu regresyon modeli ise çok değişkenli regresyon analizi olarak bilinmektedir (Korkut, 2019).

Lineer regresyon modelinde bağımlı değişkenin Y olduğu varsayılırsa, Y ile $X_1, X_2, \ldots, X_n$ ile ifade edilen n adet bağımsız değişken arasındaki ilişki için doğrusal bir denklem oluşturulur. Bu denklem tahmin edilmek istenen bağımlı değişken Y için regresyon denklemidir ve Denklem 1'deki gibi ifade edilebilmektedir (Catal vd., 2019).

$$Y = b_0 + b_1X_1 + b_2X_2 + \cdots + b_nX_n + \varepsilon \quad (1)$$

burada, $b_1, b_2, \ldots, b_n$ değerleri regresyon katsayılarını, ε ise hata terimini temsil etmektedir.

3.1.2. Yapay sinir ağları (Artificial neural networks)

Yapay sinir ağları yoğun bir şekilde birbirine bağlı yapılar ile eldeki verilerin işlendiği sistemler olarak tanımlanabilmektedir. Eğitim aşamasında önceki olayların örneklerine dayalı genelleştirme yoluyla bilgi edindikten sonra test aşamasında hiç deneyimlemediği ya da gelecekte meydana gelmesi olası bir tahmin gerçekleştirir (Gökler, 2020). Birçok sinir hücresinin birleşmesiyle yapay sinir ağları oluşur. Sinir hücresi yapay sinir ağlarında işlem yapan en küçük birimdir. Sonuç ise bir araya gelen birçok sinir hücresinin ortak sonucu olarak elde edilmektedir. Yapay sinir ağları hücrelerin birbirine bağlanmasıyla oluşur ve hücreler arasında bulunan bağlantılar ağırlıklar ile meydana getirilmektedir. Yapay sinir hücreleri arasındaki bağıntının kuvvetini ağırlıklar belirlemektedir. Yapay sinir ağının kurulabilmesi için en az birer adet giriş katmanı, gizli katman ve çıkış katmanı bulunmalıdır. Gizli katmanlar girişteki verilerin yorumlanması ve çıkışa iletilmesinde görev alır. Gizli katmanlarda işlem gören giriş verileri çıktı katmanına aktarılmaktadır. Bir yapay sinir ağı modelinde n adet girdi katmanı ve n adet çıkış katmanı olabilir (Güven, 2020).

3.1.3. Karar ormanı (Decision forest)

Açıklayıcı değişkenlerin kullanımıyla bağımlı bir değişkenin tahmini için kullanılan yöntemlerden birisi olan karar ağaçları kök, dallar ve yapraklardan oluşmaktadır. Başlangıç düğüm noktası olan kök kullanılan performans ölçütüne göre veriyi en iyi şekilde bölen değişkendir. Kök genel olarak ağaç oluşturma algoritması tarafından belirlenen kurala göre ikili dallara ayrılır ve bu dallar kendi içlerinde başka dallara ayrılabilmektedir. Tüm bu ayrılmış dallar yaprak adı verilen terminal düğümlerde son bulmaktadır. Yaprak düğümlerindeki değerler çıktıyı temsil etmektedir (Spiliotis, 2022).

Ancak basit karar ağaçları ile eğitilen modeller aşırı uyum sorunu ile sonuçlanabilmektedir. Karar Ormanı algoritması çok sayıda karar ağacı oluşturulmasını içeren bir topluluk makine öğrenmesi tekniğidir. Tek karar ağaçlarına kıyasla topluluk modelleri, daha yüksek doğruluk sağlar (Chidroop ve Moharir, 2020). Karar Ormanı modeli geliştirilirken, her karar ağacı değerlendirilir. Tüm karar ağaçlarından elde edilen sonuçlar belirlenerek yeni veriler puanlanır. Karar Ormanı Regresyonunda her karar ağacı tarafından tahmin edilen sayısal değerlerin ortalaması hesaplanarak nihai tahmin edilen sonuç elde edilmiş olur (Mohammed vd. 2023).

3.2. Nitelik seçimi (Feature selection)

Nitelik seçimi, gereksiz ve önemsiz özellikleri özellik kümesinden çıkartarak makine öğrenmesi modelinin performansını artırmayı amaçlayan bir veri ön işleme aşamasıdır (Bansal vd., 2022). Bu çalışmada, nitelik seçimi için Parçacık Sürüsü Optimizasyonu, Harris Şahinleri Optimizasyonu, Gri Kurt Optimizasyonu, Yusufçuk Optimizasyonu, Genetik Optimizasyonu ve Yerçekimi Arama Optimizasyonu algoritmalarından faydalanılmıştır.

3.2.1. Parçacık sürüsü optimizasyonu (PSO) algoritması (Particle swarm optimization algorithm)

Kuş sürülerinin davranışından esinlenen Parçacık Sürüsü Optimizasyonu algoritması 1995 yılında R Eberhart ve J Kennedy tarafından önerilmiştir (Kennedy, 2010). Parçacık sürüsü optimizasyonunda belirli sayıdaki parçacık bir fonksiyon ya da problemdeki arama boşluğuna yerleştirilir ve her bir parçacık amaç fonksiyonunu kendi bulunduğu konumda değerlendirir. Her bir parçacık sürüdeki bir veya daha fazla üye ile birlikte geçmiş en uygun konumlarını birleştirerek boşluktaki hareketlerini belirler. Tüm parçacıkların hareketi tamamlanınca bir sonraki iterasyon başlar. Nihayetinde yiyeceğe yönelmekte olan bir kuş sürüsünde olduğu gibi sürü optimum uygunluk fonksiyonuna yakın hareket etmeye başlar (Poli vd., 2007).

3.2.2. Harris şahinleri optimizasyonu (HŞO) algoritması (Harris hawk optimization algorithm)

Heidari vd. (2019) tarafından tanıtılan Harris Şahinleri Optimizasyonu (HŞO) algoritması Harris şahinlerinin doğada sürpriz saldırı adı verilen avlanma ve işbirlikçi eylemlerine dayanmaktadır. HŞO, bir grup şahinin, avın yerini takip etmek için çeşitli takip stillerini kullanarak işbirliği yaptığı popülasyona dayalı bir algoritma olarak sınıflandırılır. Burada şahin grubunun her biri bir aday çözümü avın yeri ise en uygun aday çözümü temsil etmektedir (Thaher ve Arman, 2020). Harris Şahinleri Optimizasyonu şahinlerin keşfetme ve açığa çıkarma taktiklerinden esinlenerek elde edilmiş fazlardan oluşan topluluk tabanlı degradesiz bir tekniktir. Keşif fazında takım üyeleri rastgele konumlara geçerler iki farklı strateji ile keşif aşamasını gerçekleştirebilirler. Açığa çıkarma fazında ise avların kaçış eğilimlerinden kaynaklı olarak dört farklı kovalama stratejisi önerilmiştir. Avın kaçışının başarılı ya da başarısız olması durumuna göre sert ya da yumuşak kuşatma stratejisi belirlenmektedir. Avın kalan enerjisi de göz önünde bulundurularak yumuşak ya da sert kuşatma döngüleri devam eder ve avın yorgun düşmesi ile hedefe ulaşılır (Heidari vd., 2019).

3.2.3. Gri kurt optimizasyonu (GKO) algoritması (Grey wolf optimization algorithm)

Gri Kurt Algoritması Mirjalili vd. (2014) tarafından kurt sürülerinin avlarını yakalama süreçlerinin canlandırmasını yansıtacak şekilde sunulmuştur. Bu süreçler avlarını takip etme, çevreleme ve ava saldırma aşamalarını içermektedir. Algoritma nispeten az kontrol parametreleri içermektedir ve uygulaması kolaydır. Gri kurt sürülerindeki sosyal hiyerarşiyi yansıtacak şekilde uygunluk değerini elde etmek amacıyla bir çözüm, alt çözüm ve de üçüncü bir çözüm olacak bulunmaktadır. Algoritmada bu çözümler sırasıyla alfa, beta ve delta olarak adlandırılmaktadır (Zeng vd., 2022).

3.2.4. Yusufçuk optimizasyonu (YO) algoritması (Dragon fly optimization algorithm)

Yusufçuk Optimizasyonu, yusufçukların eşsiz avlanma ve göç etme davranışlarından esinlenerek oluşturulmuştur. Avlanan sürü davranışı diğer bilinen adıyla statik sürü davranışı, küçük sürü gruplarının ani adım değişiklikleri ve yerel hareketleriyle tanımlanmaktadır. Göçsel sürü davranışı ise aynı zamanda dinamik sürü olarak bilinmektedir ve tek bir yönde giden yüksek sayıdaki yusufçuğu ifade eder. Statik ve dinamik kavramları Yusufçuk Algoritmasının açığa çıkarma ve keşif kapasitelerini belirtmektedir. Ayrışma, dizilme, birlikte durma, dikkat dağıtma ve yiyeceğe çekim gibi ögeler içeren davranış tanımlanmaktadır. Her bir yusufçuk arama boşluğundaki bir çözüme karşılık gelir ve bu beş farklı öge ile sürü hareketi belirlenir (KS ve Murugan, 2017).

3.2.5. Genetik optimizasyonu (GO) algoritması (Genetic optimization algorithm)

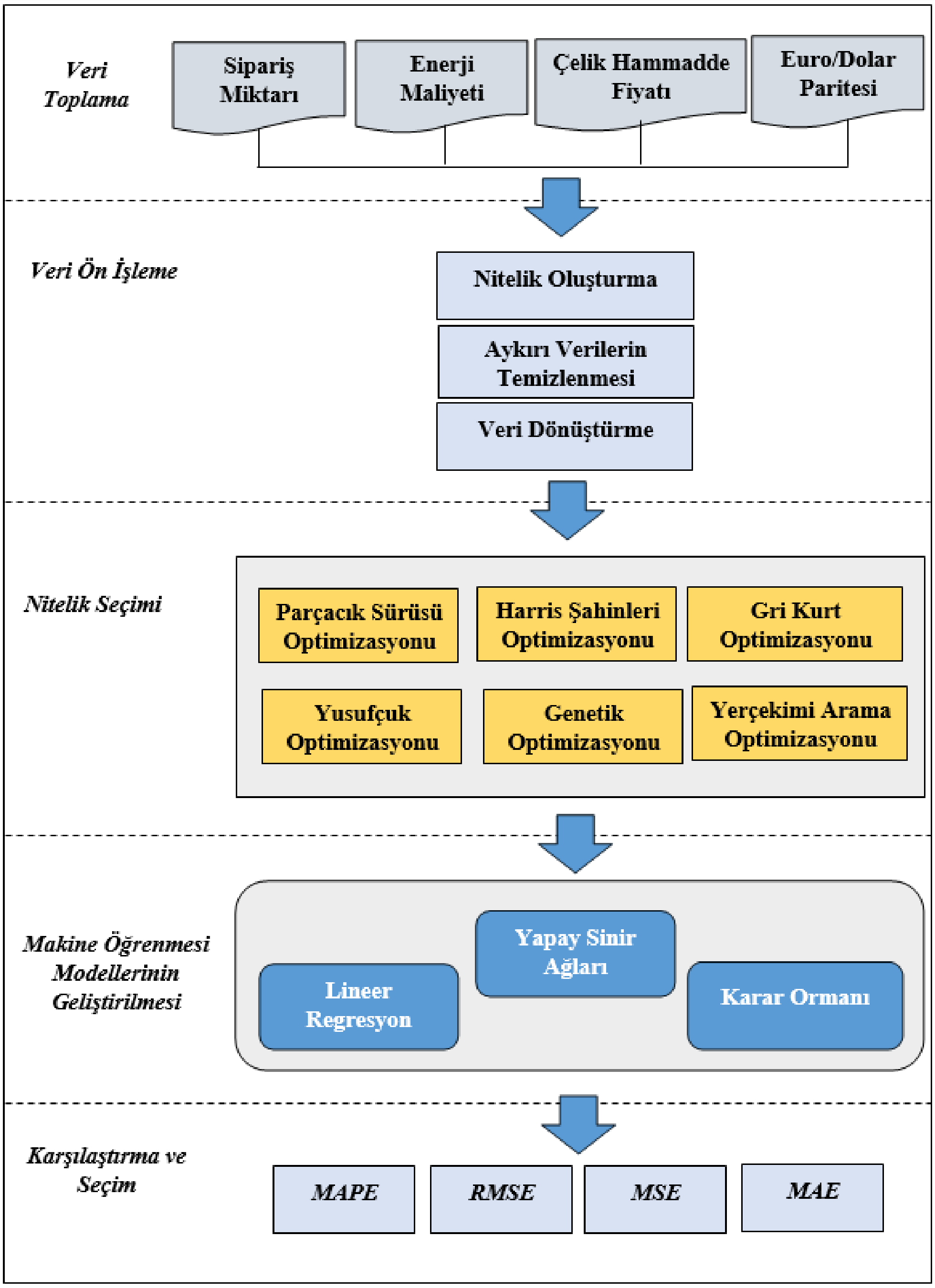
Genetik Optimizasyon algoritmaları doğal biyolojik süreçlerin adapte edilmesi ile elde edilen hesaplama yöntemlerden biridir. Genetik optimizasyonu çoklu çözüm vektörlerini aynı anda sürekli değerlendirmesi ayrıcalığıyla geleneksel yöntemlerden daha avantajlıdır. Genetik algoritmalar rastgele oluşturulmuş ve topluluğa karşılık gelen karar vektörlerinden oluşmaktadır. Her bir karar vektörü grup pürüzlülük değerleri ve talep ayarlama faktörleri içeren karar değişkenleri setinden oluşmaktadır. Her bir karar vektörü ikili sayılardan oluşmaktadır ve karar değişkeni değeri ikili sayı sitemine göre çevrilmektedir. İkili dizi setinin üst ve alt limitleri karar değişkeninin muhtemel en küçük varyasyonunu kontrol eder. Karar değişkeni değerleri ikili sisteme dönüştürülür ve ikili sistemden tekrar eski haline dönüştürülebilmektedir. İkili sistem kodlaması sayesinde karar vektörü 0 ve 1'lerden oluşan ikili diziye indirgenebilmektedir. Elde bulunan karar vektörlerinden uygunluk fonksiyonuna göre alt karar vektörleri seti seçilir. Bir karar vektörü popülasyonu seçildiğinde çaprazlama ya da mutasyon operasyonlarıyla yeni bir karar vektörü popülasyonu oluşturulur (Lingireddy ve Ormsbee, 2002).

3.2.6. Yerçekimi arama optimizasyonu (YAO) algoritması (Gravitational optimization algorithm)

Yerçekimi Arama Optimizasyon Algoritması Newton'un kütle çekim teorisinden esinlenerek ortaya çıkarılmıştır. Kütle çekim teorisinde kuvvet kütle ile doğru orantı ve uzaklığın karesi ile ters orantı olacak şekilde tanımlanmaktadır. Yerçekimi arama algoritmasında tanımlanmış olan parçacıklar pozisyon, kütle, aktif ve pasif yerçekimi olmak üzere dört farklı özellik taşımaktadır. Problemin çözümü parçacığın pozisyonu ile sağlanmaktadır ve kütleler uygunluk fonksiyonuna göre belirlenebilmektedir. Her bir kütle çekim kanunu ve hareket kanunu olmak üzere iki kanuna tabidir (Yadav ve Deep, 2013).

4. Uygulama (Implementation)

Bu çalışmada, nitelik seçimi yöntemleri ve makine öğrenmesi algoritmalarının hibrit kullanımı ile talep tahmin modelleri geliştirilmiştir. Çalışmanın metodolojisi Şekil 1'de verilmektedir. Şekilde görüldüğü gibi çalışma 5 temel adımdan oluşmaktadır. Bu adımlar veri toplama, veri ön işleme, nitelik seçimi, tahmin modellerinin geliştirilmesi ve modellerin karşılaştırılması ve seçimidir.



Şekil 1. Çalışmanın metodolojisi (Methodology of the study)

4.1. Veri toplama (Data collection)

Bu çalışmada talep tahmini modellerinin geliştirilmesi için kullanılacak olan veriler ana bileşeni çelik olan ürünlerin taleplerini doğrudan ya da dolaylı olarak etkileyen parametreler arasından seçilmiştir. İlk değişken olarak çelik sektöründe ürün satışı yapan büyük ölçekli uluslararası bir şirketten gerçek satış verileri elde edilmiştir. Bu veriler Nisan 2016 ve Ağustos 2023 tarihleri arasında 4 ürüne ait 89 aylık satış miktarı verilerinden oluşmaktadır. Ürünlerin seçimi aşamasında, ürün yelpazesinde satış miktarı en fazla olan ve sipariş dalgalanması nispeten az olan ürünlerin seçilmesine dikkat edilmiştir. Ürünler çalışmada A ürünü, B ürünü, C ürünü ve D ürünü olarak isimlendirilmiştir.

Ürünlerin imalatında kullanılan hammaddelerin uluslararası düzeyde dolar bazında işlem görmesi ve satışlarının euro bazında yapılması nedeniyle euro/dolar paritesi bağımsız değişkenlerden biri olarak seçilmiştir. Bu bağımsız değişkene ait veriler anlık finansal verilerin sağlandığı bir web sitesinden elde edilmiştir. Diğer taraftan, ürünlerin maliyetinin yaklaşık olarak %80 kadarının hammadde kaynaklı olması ve ürünlerdeki ana bileşenin çelik olması nedeniyle çelik fiyatı, veri setine bağımsız değişken olarak eklenmiştir. Çelik fiyatı verileri ise satış verileri gibi firma veritabanından elde edilmiştir. Ürünlerin maliyetinde hammadde ve euro bazında yapılan makine yatırım bedelinden sonra en önemli maliyet kaleminin enerji maliyeti olması sonucu bir diğer bağımsız değişken olarak enerji maliyeti kullanılmıştır.

4.2. Veri ön işleme (Data preprocessing)

Çalışmanın bu aşamasında nitelik oluşturma, aykırı verilerin temizlenmesi, normalizasyon ve veri dönüştürme olmak üzere dört veri ön işleme süreci gerçekleştirilmiştir.

4.2.1. Nitelik oluşturma (Feature generation)

Bu adımda, talep tahmini modellerinde kullanılmak üzere sipariş miktarı, enerji maliyeti, çelik hammadde fiyatı ve euro/dolar paritesi değişkenlerine ek olarak kayan pencereler, kayan pencere istatistikleri, birinci derece farkı nitelik oluşturma işlemleri yardımı ile bir, iki ve üç ay önceki sipariş miktarları, son üç ayın siparişlerinin ortalaması, enerji maliyetindeki değişim, çelik fiyatındaki değişim ve sipariş miktarındaki değişim olmak üzere 7 farklı değişken elde edilmiştir.

Kayan Pencere: Gecikme değişkenleri, geçmişte olanların geleceğe ilişkin bir tür içsel bilgiyi etkileyebileceği veya içerebileceği varsayımına dayanarak oluşturuldukları için yararlı olduğu düşünülen önceki zaman adımlarındaki değerlerdir. Veri setine gecikme değişkenleri ekleme işlemine kayan pencere yöntemi adı verilmektedir (Lazzeri, 2020). Bu çalışmada her bir ürünün sipariş miktarı niteliğine 3 pencere genişliğine sahip kayan pencereler yöntemi uygulanarak 1 ay önceki sipariş miktarı, 2 ay önceki sipariş miktarı ve 3 ay önceki sipariş miktarı değişkenleri elde edilmiştir.

Kayan Pencereler İstatistikleri: Bir zaman serisi veri setinde kayan pencereler istatistikleri oluşturmanın temel amacı, örneğin kendisini ve örnekten önceki ve sonraki belirli sayıda örneği içeren bir aralık tanımlayarak belirli bir veri örneğinden elde edilen değerlere ilişkin istatistikleri hesaplamaktır (Lazzeri, 2020). Bu çalışmada, sipariş miktarı niteliğine en popüler kayan pencereler istatistiklerinden olan hareketli ortalama istatistiği uygulanarak son üç ayın siparişlerinin ortalaması niteliği elde edilmiştir. Kayan pencere aralığı örneğin kendisinden önceki 3 ay olarak seçilmiştir.

Birinci Derece Farkı: Bir zaman serisi niteliğinin birinci derece farkı, zaman serisi niteliğinin her birim zaman adımı arasındaki değişim oranı olarak tanımlanabilmektedir. Birinci derece fark ile birlikte, bir zaman serisinin eğimi belirlenebilir, belirli aykırı değerlerin varlığı gibi zaman serisi hakkında ek bilgi edinilebilir. X={x_1, x_2, ..., x_t} olarak verilen bir zaman serisi değişkeni Denklem (2) kullanılarak birinci derece farkına dönüştürülebilir (Tan vd., 2022).

$$X' = \{x_t - x_{t-1}\} \quad (2)$$

Bu çalışmada enerji maliyeti, çelik hammadde fiyatı ve sipariş miktarı niteliklerine birinci derece fark dönüşümü uygulanarak enerji maliyetindeki değişim, çelik fiyatındaki değişim ve sipariş miktarındaki değişim değişkenleri elde edilmiştir.

A ürünü, B ürünü, C ürünü ve D ürünü için elde edilen veri setlerindeki niteliklere ait istatistiksel özellikler Tablo 1'de sunulmaktadır. Burada, enerji maliyeti, çelik hammadde fiyatı, euro/dolar paritesi, bir, iki ve üç ay önceki sipariş miktarları, son üç ayın siparişlerinin ortalaması, enerji maliyetindeki değişim, çelik fiyatındaki değişim ve sipariş miktarındaki değişim bağımsız değişkenler iken sipariş miktarı bağımlı değişkendir. Euro/dolar paritesi, enerji maliyeti, enerji maliyetindeki değişim, çelik fiyatı ve çelik fiyatındaki değişim değişkenlerindeki değerler 4 veri seti için aynı değerleri almaktadır. Sipariş miktarı, 1 ay önceki sipariş miktarı, 2 ay önceki sipariş miktarı, 3 ay önceki sipariş miktarı, son 3 ayın siparişlerinin ortalaması ve sipariş miktarındaki değişim değişkenleri değerleri ise ürünlere göre değişkenlik göstermektedir. Tabloda bulunan benzersiz veriler sütunu bir nitelikte bulunan birbirinden farklı değer sayısını belirtmektedir.

4.2.2. Aykırı verilerin temizlenmesi (Removing outliers)

Modellerin geliştirilmesinde kullanılacak olan veri setlerinde, veri setlerindeki gürültülerin azaltılması ve geliştirilecek modellerin uç değerlerden olumsuz etkilenmesinin önlenmesi amacıyla aykırı değer temizleme işlemi gerçekleştirilmiştir. Aykırı değerlerin temizlenmesinde yüzdelik dilim kullanılarak üst tepe ve alt tepe değerlerin kırpılarak niteliğin ortalama değeri ile değiştirilmesi yöntemi temel alınmıştır. Üst eşik yüzdelik dilim 95, alt eşik yüzdelik dilim ise 5 olarak belirlenmiştir.

4.2.3. Veri dönüştürme (Data transformation)

Tablo 1 incelendiğinde 4 veri setinde kullanılan ortak niteliklerden enerji maliyeti ve çelik fiyatı değişkenlerinin oldukça farklı değer aralıklarında değerler aldığı görülmektedir. Modellerin performansını iyileştirmek amacıyla bu niteliklerin ortak bir ölçekte değerler almalarını sağlamak için bu çalışmada veri dönüştürme yöntemlerinden birisi olan MinMax normalizasyon işlemi uygulanmıştır. En yaygın kullanılan normalizasyon yöntemlerinden birisi olan MinMax normalizasyon yöntemi değerleri oldukça büyük aralıkta değişen verileri daha küçük aralığa dönüştürür (Özçelik vd., 2021). X normalize edilmek istenen nitelik değeri, X_n normalize edilmiş yeni değer, X_{min} niteliğin minimum değeri ve X_{maks} niteliğin maksimum değeri olmak üzere MinMax normalizasyon işleminin formülasyonu Denklem (3)'te verilmektedir.

Tablo 1. A ürünü, B ürünü, C ürünü ve D ürünü veri setlerinin istatistiksel özellikleri (Statistical properties of product A, product B, product C and product D datasets)

	Nitelikler	Ortalama	Medyan	Min	Maks	Standart Sapma	Benzersiz Veriler	Eksik Veriler
Ortak Nitelikler	Eur/Dolar paritesi	1.132	1.13	0.98	1.24	0.056	25	0
	Enerji maliyeti	98.52	88	56	192	31.8	51	0
	Enerji maliyetindeki değişim	0.21	-2	-31	71	13.3	32	0
	Çelik fiyatı	679.1	591	332	1520	269.6	67	0
	Çelik fiyatındaki değişim	6.53	0	-259	264	72.04	44	0
A Ürünü	Sipariş miktarı	16322	14560	3952	39728	8170	67	0
	1 ay önceki sipariş miktarı	6479	14768	3952	39728	8124	67	0
	2 ay önceki sipariş miktarı	16596	15184	3952	39728	8102	67	0
	3 ay önceki sipariş miktarı	16668	15184	3952	39728	8054	67	0
	Son 3 ay sipariş ortalaması	16581	16085	7259	29328	6082	75	0
	Sipariş miktarındaki değişim	116.5	0	-24752	25168	9113	69	0
B Ürünü	Sipariş miktarı	1997	1841	479	7614	953	88	0
	1 ay önceki sipariş miktarı	1993	1841	479	7614	961	88	0
	2 ay önceki sipariş miktarı	2004	1841	479	7614	953	88	0
	3 ay önceki sipariş miktarı	2015	1850	479	7614	944	88	0
	Son 3 ay sipariş ortalaması	2004	1936	696	3880	747	89	0
	Sipariş miktarındaki değişim	-11	29	-5773	5732	1032	86	0
C Ürünü	Sipariş miktarı	12344	11756	1249	23913	4679	89	0
	1 ay önceki sipariş miktarı	12398	11901	1249	23913	4647	89	0
	2 ay önceki sipariş miktarı	12466	11945	1249	23913	4597	89	0
	3 ay önceki sipariş miktarı	12408	11945	1249	23913	4664	89	0
	Son 3 ay sipariş ortalaması	12424	12787	6969	18845	2474	88	0
	Sipariş miktarındaki değişim	68	-142	-20411	17793	7240	89	0
D Ürünü	Sipariş miktarı	1819	1704	561	9080	1000	88	0
	1 ay önceki sipariş miktarı	1831	1711	561	9080	995	88	0
	2 ay önceki sipariş miktarı	1835	1711	561	9080	994	88	0
	3 ay önceki sipariş miktarı	1845	1723	561	9080	990	88	0
	Son 3 ay sipariş ortalaması	1837	1779	677	5051	759	89	0
	Sipariş miktarındaki değişim	3.51	-35	-6306	6988	1115	84	0

$$X_n = \frac{X - X_{min}}{X_{maks} - X_{min}} \tag{3}$$

Veriler arasındaki dengesizliklerin ve çarpıklıkların giderilmesinde kullanılan yöntemlerden birisi de doğal logaritma (Ln) dönüşümdür. Çok geniş veri aralıklarında daha küçük olan değerler büyük değerler tarafından bastırılabilir. Ln dönüşümü aykırı değerlerin dağılımdaki ağırlığını azaltır. Bu sayede dağılımda simetri tekrar elde edilmiş olur (Sauro ve Lewis, 2016). Modelin eğitiminde kullanılan gerçek sipariş miktarı değerlerinin aralığı çok geniş olduğundan verilerde çarpık dağılımın giderilerek normal dağılıma yakın bir dağılım izlemesi için logaritmik dönüşüm gerçekleştirilmiştir.

Eğitim veri seti ile tahmin veri seti arasındaki sabit ve dinamik durumun dengelenmesi ve modelin daha doğru genelleme yapılabilmesi için veri öteleme yöntemleri tercih edilmektedir (Huyen, 2022). Bu çalışmada tahmin modellerinin gerçek değerlere daha yakın tahminler sağlayabilmesi için matematiksel öteleme işlemi gerçekleştirilmiştir. Eğitim veri setinde Ln dönüşümü gerçekleştirilmiş gerçek sipariş miktarı değerlerine, niteliğin ortalama sapma değerine yakın bir değer çıkartılması işlemi uygulanmıştır. Bu değer A ürünü için 0.15, B ve C ürünleri için 0.30, D ürünü için ise 0.38'dir.

4.3. Nitelik seçimi (Feature selection)

Bu çalışmada, ürün veri setlerindeki sipariş miktarı değişkeninin tahmin edilmesinde en ilgili özelliklerin belirlenmesi için Parçacık Sürüsü Optimizasyonu, Harris Şahinleri Optimizasyonu, Gri Kurt Optimizasyonu, Yusufçuk Optimizasyonu, Genetik Optimizasyonu ve Yerçekimi Arama Optimizasyonu algoritmalarından faydalanılmıştır. Algoritmaların tüm nitelikler üzerinde uygulanması adımları Spyder 3.1 geliştirme ortamında Python programlama dili ile açık kaynaklı 'zoofs' kütüphanesi yardımıyla yürütülmüştür. Uygulamada optimizasyon algoritmaları için kullanılan parametreler ve değerleri Tablo 2'de sunulmaktadır.

Optimizasyon algoritmalarının uygulanmasında 'LightGBM' Python kütüphanesinden (LightGBM, 2023) yararlanılarak LightGBM makine öğrenmesi regresyon modeli kullanılmıştır. Model parametrelerinden iterasyon sayısı 100, öğrenme oranı 0.1 ve yaprak sayısı 31 olarak belirlenmiştir. Uygulama sonucunda optimizasyon yöntemlerinin sunduğu nitelik seçimleri Tablo 3'te gösterilmiştir. Niteliğin ilgili optimizasyon algoritması çıktısında yer alması durumu 'X' işareti ile belirtilmiştir.

Tablo 2. Nitelik seçimi algoritmaları için kullanılan parametreler (Parameters used for feature selection algorithms)

Nitelik Seçimi Algoritması	Parametre	Değer
Parçacık Sürüsü Optimizasyonu	iterasyon sayısı	1000
	popülasyon büyüklüğü	20
	ilk ivme katsayısı	2.0
	ikinci ivme katsayısı	2.0
	ağırlık parametresi	0.9
	amaç fonksiyonu	Hata kareleri ortalaması performans ölçütünün minimizasyonu
Harris Şahinleri Optimizasyonu	iterasyon sayısı	250
	popülasyon büyüklüğü	20
	amaç fonksiyonu	Hata kareleri ortalaması performans ölçütünün minimizasyonu
	beta	0.5
Gri Kurt Optimizasyonu	iterasyon sayısı	1000
	popülasyon büyüklüğü	20
	metot	1
	amaç fonksiyonu	Hata kareleri ortalaması performans ölçütünün minimizasyonu
Yusufçuk Optimizasyonu	iterasyon sayısı	1000
	popülasyon büyüklüğü	20
	metot	ikinci dereceden
	amaç fonksiyonu	Hata kareleri ortalaması performans ölçütünün minimizasyonu
Genetik Optimizasyonu	iterasyon sayısı	1000
	popülasyon büyüklüğü	20
	seçici baskı	2
	elitizm	3
	mutasyon oranı	0.05
	amaç fonksiyonu	Hata kareleri ortalaması performans ölçütünün minimizasyonu
Yerçekimi Arama Optimizasyonu	iterasyon sayısı	15
	popülasyon büyüklüğü	50
	yerçekimi kuvveti sabiti	100
	mesafe sabiti	0.5
	amaç fonksiyonu	Hata kareleri ortalaması performans ölçütünün minimizasyonu

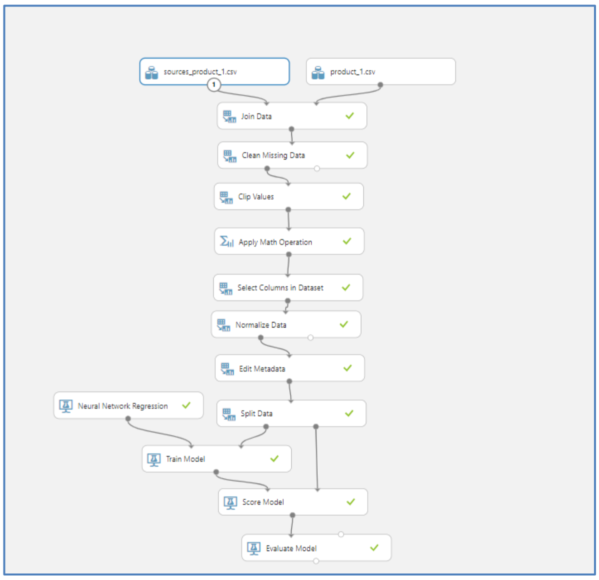
Tablo 3. Optimizasyon yöntemleri tarafından seçilen nitelikler (Features selected by optimization methods)

Nitelikler	**PSO**	**HŞO**	**GKO**	**YO**	**GO**	**YAO**
Euro/Dolar paritesi	X	X	X	X	X	
Enerji maliyeti	X	X		X	X	X
Enerji maliyetindeki değişim		X			X	
Çelik fiyatı		X	X		X	X
Çelik fiyatındaki değişim	X	X	X	X	X	X
1 ay önceki sipariş miktarı	X	X	X	X		
2 ay önceki sipariş miktarı	X	X	X	X		X
3 ay önceki sipariş miktarı			X			X
Son 3 ay sipariş ortalaması		X	X		X	X
Sipariş miktarındaki değişim		X	X		X	X

Tablo 3 incelendiğinde, PSO, HŞO, GKO, YO, GO ve YAO algoritmaları tarafından sırasıyla 5, 9, 8, 5, 7 ve 7 adet bağımsız niteliğin seçildiği görülmektedir. Çelik fiyatındaki değişim bağımsız değişkeni tüm nitelik seçimi optimizasyon algoritmaları tarafından sipariş miktarının tahmin edilmesinde etkili bir değişken olarak bulunmuştur. Genel olarak değerlendirildiğinde ise Euro/Dolar paritesi, enerji maliyeti 1 ay önceki sipariş miktarı, 2 ay önceki sipariş miktarı değişkenleri 5 optimizasyon algoritması tarafından etkili bulunmuştur.

4.4. Makine öğrenmesi modellerinin geliştirilmesi (Development of machine learning models)

Çalışmada, A, B, C ve D olarak isimlendirilen 4 farklı ürün veri seti üzerinde tüm bağımsız değişkenlerin kullanılması ile ve Parçacık Sürüsü Optimizasyonu, Harris Şahinleri Optimizasyonu, Gri Kurt Optimizasyonu, Yusufçuk Optimizasyonu, Genetik Optimizasyonu ve Yerçekimi Arama Optimizasyonu algoritmaları ile seçilen niteliklerin kullanılması ile olmak üzere Lineer Regresyon, Yapay Sinir Ağları ve Karar Ormanı algoritmaları kullanılarak 84 (4 ürün X 7 nitelik seçimi durumu X 3 algoritma) adet talep tahmini modeli geliştirilmiştir. Tahmin modellerinin geliştirilmesinde ve test edilmesinde Microsoft Azure Machine Learning (ML) Studio ortamından yararlanılmıştır. Azure ML Studio ortamında tahmin modeline ait mimari yapı Şekil 2'de gösterilmiştir.



Şekil 2. Azure ML Studio tahmin modeli mimarisi (Azure ML Studio prediction model architecture)

Muraina (2022), makine öğrenmesi algoritmalarının tahmin doğruluğunu artırmak için en uygun veri seti bölme oranını belirlemek amacıyla gerçekleştirdiği çalışmada en yüksek doğruluğu sağlayan oranın %90 (eğitim veri seti) - %10 (test veri seti) olduğu sonucuna varmıştır. Bu çalışmada da, 4 ürün veri setindeki verilerin %90'ı (80 aylık veriler) eğitim, %10'u (9 aylık veriler) test veri seti olarak kullanılmıştır. Her bir veri seti için deneme-yanılma yöntemi ile belirlenen algoritma eğitim parametreleri ve değerleri Tablo 4'te gösterilmiştir.

4.5. Performans değerlendirme (Performance evaluation)

Literatürde talep tahmin çalışmalarında geliştirilen modellerin değerlendirilmesi ve karşılaştırılması amacıyla kullanılan birçok performans değerlendirme kriteri bulunmaktadır. Bu çalışmada modellerin performanslarının değerlendirilmesinde ortalama mutlak yüzde hata (MAPE), hata kareleri ortalaması (MSE), hata kareleri ortalamasının karekökü (RMSE), ortalama mutlak hata (MAE) ve tahmin doğruluğu ölçütleri kullanılmıştır. Performans ölçütlerinin formülleri sırasıyla Denklem 4-8'de verilmektedir (Yaşar vd., 2021; Demircioglu Diren vd., 2020; Feizabadi, 2022; Chicco vd., 2021; Kacar, 2024).

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{Y_i-\hat{Y}_i}{Y_i}\right| \tag{3}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i-\hat{Y}_i\right)^2 \tag{4}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i-\hat{Y}_i)^2} \tag{5}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}Y_i-\hat{Y}_i \tag{6}$$

$$\text{Tahmin doğruluğu}(\%) = 1 - \text{MAPE} \tag{7}$$

Burada, n, test veri setindeki toplam veri noktası sayısıdır, Y_i i. örneğin satış miktarının gerçek değeridir ve $\hat{Y}_i$ i. örneğin satış miktarının tahmin edilen değeridir.

4.6. Deneysel sonuçlar (Experimental results)

Tablo 5'te A ürünü için geliştirilen talep tahmini modellerinin performans ölçütü değerleri verilmektedir. Deneysel sonuçlara göre, tüm performans ölçütü değerlerinde en düşük değerleri sağlayan model tüm bağımsız nitelikler ile geliştirilen Yapay Sinir Ağları modeli olmuştur. Modelin test edilmesi ile elde edilen MAPE, RMSE, MSE ve MAE değerleri sırası ile %6.4, 609, 370505 ve 502'dir. Lineer Regresyon Algoritması ile geliştirilen modeller incelendiğinde, PSO, HŞO, GKO, YO, GO, YAO algoritmaları olmak üzere tüm nitelik seçimi yöntemlerinin uygulanmasıyla elde edilen modeller tüm niteliklerin kullanılması ile geliştirilen modellerden daha iyi performans göstermişlerdir. MAPE değerlerine göre, PSO algoritması %4.1, HŞO algoritması %1, GKO algoritması %6.2, YO algoritması %4, GO algoritması %0.4, YAO algoritması ise %6.5 oranında iyileştirme sağlamıştır. Karar Ormanı algoritması ile geliştirilen modellerin sonuçları analiz edildiğinde ise GO nitelik seçimi yönteminin tüm performans ölçütü değerleri açısından modellerin performanslarını geliştirdiği anlaşılmaktadır. Bu iyileştirme MAPE değerinin %10.9'dan %8.6'ya, RMSE değerinin 955'ten 819'a, MSE değerinin 912785'ten 670423'e, MAE değerinin ise 831'den 693'e düşmesiyle elde edilmiştir.

Tablo 4. Modellerde kullanılan parametreler ve değerleri (Parameters and values used in the models)

Algoritma	Parametre	A Ürünü	B Ürünü	C Ürünü	D Ürünü
Lineer Regresyon	Çözüm metodu	En küçük kareler	En küçük kareler	En küçük kareler	En küçük kareler
	L2 düzenlileştirme ağırlığı	0.001	0.001	0.001	0.001
Yapay Sinir Ağları	Gizli düğüm noktası sayısı	8	8	8	8
	Öğrenme oranı	0.002	0.001	0.001	0.003
	Öğrenme iterasyon sayısı	12000	12000	30000	12000
	İlk öğrenme ağırlığı	0.01	0.01	0.05	0.01
	Momentum değeri	0.2	0.5	0.4	0.2
	Normalize yöntemi	Min-Maks	Min-Maks	Min-Maks	Min-Maks
Karar Ormanı	Yeniden örnekleme yöntemi	Torbalama	Torbalama	Torbalama	Torbalama
	Karar ağacı sayısı	8	8	8	8
	Karar ağaçlarının maksimum derinliği	32	32	32	32
	Düğüm başına rastgele bölünme sayısı	298	256	256	128
	Yaprak düğümü başına minimum örnek sayısı	1	1	1	2

B ürünü için geliştirilen modellerin test edilmesi ile elde edilen performans ölçütü değerleri Tablo 6'da sunulmuştur. Değerler incelendiğinde en düşük MAPE ve MAE sonuçlarını sırasıyla %5.3 ve 66 değerleri ile HŞO algoritması ile seçilen nitelikler üzerinde eğitilen Lineer Regresyon modelinin verdiği görülmektedir. En düşük RMSE ve MSE sonuçlarını ise 92 ve 8523 değerleri ile YAO algoritması ile seçilen nitelikler üzerinde eğitilen Lineer Regresyon modeli sağlamıştır. HŞO nitelik seçimi yönteminin uygulanmasının Yapay Sinir Ağları ve Karar Ormanı algoritmaları ile geliştirilen modellerin tüm performans ölçütleri açısından performansını geliştirdiği anlaşılmaktadır.

C ürünü için geliştirilen modellerin performans ölçütü değerleri Tablo 7'de görülmektedir. Modeller arasında MAPE, RMSE, MSE ve MAE olmak üzere dört performans ölçütü açısından sırasıyla %9.7, 949, 900005 ve 747 değerleri ile en iyi tahmin sonuçlarını sağlayan model GO nitelik seçimi yöntemi ile seçilen nitelikler üzerinde eğitilen Yapay Sinir Ağları modeli olmuştur. Lineer Regresyon algoritması ile eğitilen modeller incelendiğinde, GKO nitelik seçimi yöntemi hariç tüm nitelik seçimi yöntemlerinin tüm performans kriterleri açısından daha düşük değerler elde edilmesini sağladığı görülmektedir. Karar Ormanı algoritması ile eğitilen modellerin sonuçları ise modellerin eğitiminde tüm niteliklerin kullanımının en düşük performans ölçütü değerlerini verdiğini göstermektedir.

Tablo 5. A ürünü için geliştirilen modellerin performans ölçütü değerleri (Performance criterion values of the models developed for product A)

Model	Nitelik	MAPE	RMSE	MSE	MAE
Lineer Regresyon	Tüm Nitelikler	%17.1	1629	2652351	1420
	PSO	%13	1223	1496453	997
	HŞO	%16.1	1503	2259167	1314
	GKO	%10.9	1044	1089916	892
	YO	%13.1	1225	1500479	1004
	GO	%16.7	1561	2437814	1363
	YAO	%10.6	1148	1317273	912
Yapay Sinir Ağları	Tüm Nitelikler	**%6.4**	**609**	**370505**	**502**
	PSO	%13	1315	1727909	1087
	HŞO	%10.5	856	732871	825
	GKO	%12.7	1065	1134921	984
	YO	%13	1315	1727909	1087
	GO	%7.8	905	818922	646
	YAO	%13.9	1180	1393143	1065
Karar Ormanı	Tüm Nitelikler	%10.9	955	912785	831
	PSO	%15.7	1501	2252368	1203
	HŞO	%13.2	1223	1495537	982
	GKO	%10.2	1030	1059897	845
	YO	%15.7	1501	2253574	1204
	GO	%8.6	819	670423	693
	YAO	%11.7	1067	1138510	876

Tablo 6. B ürünü için geliştirilen modellerin performans ölçütü değerleri (Performance criterion values of the models developed for product B)

Model	Nitelik	MAPE	RMSE	MSE	MAE
Lineer Regresyon	Tüm Nitelikler	%5.8	94	8861	70
	PSO	%14	189	35526	170
	HŞO	**%5.3**	99	9699	**66**
	GKO	%6.2	101	10097	74
	YO	%14	189	35526	170
	GO	%6.7	100	9941	78
	YAO	%5.9	**92**	**8523**	70
Yapay Sinir Ağları	Tüm Nitelikler	%7.4	99	9787	85
	PSO	%15.6	204	41616	187
	HŞO	%6.2	93	8692	74
	GKO	%7.6	110	12089	89
	YO	%15.6	204	41616	187
	GO	%7.2	97	9455	81
	YAO	%8.0	114	12961	93
Karar Ormanı	Tüm Nitelikler	%8.1	112	12486	95
	PSO	%16.4	213	45328	196
	HŞO	%6.1	98	9657	74
	GKO	%8.6	120	14327	100
	YO	%16.4	213	45328	196
	GO	%9.3	129	16604	111
	YAO	%9.9	140	19624	117

Tablo 8’de D ürünü veri seti üzerinde eğitilip test edilen modeller için elde edilen performans değerlendirme kriterleri değerleri verilmiştir. MAPE ölçütü açısından en iyi performansı %8.5 değeri ile GO nitelik seçimi algoritması ve Karar Ormanı algoritmasının hibrit kullanımı ile geliştirilen modelin sağladığı görülmüştür. En düşük RMSE, MSE ve MAE ölçütü değerlerini ise 141, 19736 ve 88 değerleri ile tüm nitelikler üzerinde eğitilen Yapay Sinir Ağları modeli vermiştir.

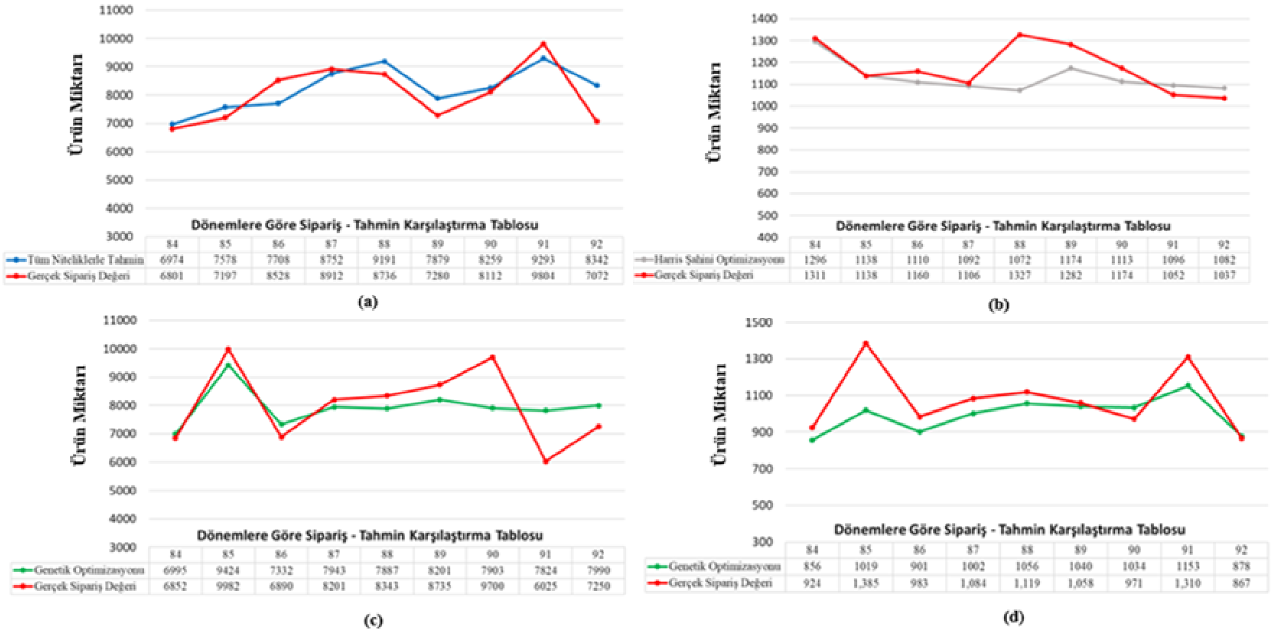
Şekil 3’te A, B, C ve D olmak üzere 4 ürün için test veri setinde bulunan gerçek siparişleri ile MAPE ölçütü açısından en iyi performansı sağlayan makine öğrenmesi modellerinin tahmin sonuçlarının karşılaştırmalı değerleri sunulmaktadır. Burada, A ürünü için en iyi model tüm nitelikler üzerinde eğitilmiş Yapay Sinir Ağları modeli (Şekil 3(a)), B ürünü için HO nitelik seçimi yöntemi ile seçilen nitelikler üzerinde eğitilen Lineer Regresyon modeli (Şekil 3(b)), C ürünü için GO nitelik seçimi yöntemi ile seçilen nitelikler üzerinde eğitilen Yapay Sinir Ağları modeli (Şekil 3(c)), D ürünü için ise GO nitelik seçimi yöntemi ile seçilen nitelikler üzerinde eğitilen Karar Ormanı modelidir (Şekil 3(d)). Değerler incelendiğinde, 4 modelin de test veri seti üzerindeki gözlemler için oldukça düşük tahmin hataları ile sonuçlandıkları görülmektedir.

Tablo 7. C ürünü için geliştirilen modellerin performans ölçütü değerleri (Performance criterion values of the models developed for product C)

Model	Nitelik	MAPE	RMSE	MSE	MAE
Lineer Regresyon	Tüm Nitelikler	%19.1	1666	2776053	1404
	PSO	%14.6	1288	1659965	1019
	HŞO	%17.7	1493	2228261	1295
	GKO	%21.1	1822	3318907	1532
	YO	%14.6	1288	1659965	1019
	GO	%17.6	1468	2153954	1270
	YAO	%14.8	1399	1957631	1037
Yapay Sinir Ağları	Tüm Nitelikler	%10.3	1023	1045607	776
	PSO	%10.3	1061	1125004	842
	HŞO	%14.8	1410	1986669	1097
	GKO	%11.2	1301	1691307	847
	YO	%10.3	1061	1125004	842
	GO	**%9.7**	**949**	**900005**	**747**
	YAO	%19.1	1746	3049300	1340
Karar Ormanı	Tüm Nitelikler	%11.6	1115	1244014	842
	PSO	%18.4	1722	2965791	1322
	HŞO	%16.9	1438	2067333	1232
	GKO	%14.1	1381	1906241	1007
	YO	%18.4	1721	2960334	1321
	GO	%16.6	1649	2720632	1244
	YAO	%18.9	1630	2656749	1368

Tablo 8. D ürünü için geliştirilen modellerin performans ölçütü değerleri (Performance criterion values of the models developed for product D)

Model	Nitelik	MAPE	RMSE	MSE	MAE
Lineer Regresyon	Tüm Nitelikler	%18.1	259	66805	207
	PSO	%23.5	321	102987	265
	HŞO	%18.1	260	67497	208
	GKO	%16.9	252	63530	194
	YO	%23.5	321	102987	265
	GO	%15.1	213	45350	171
	YAO	%14.3	218	47455	164
Yapay Sinir Ağları	Tüm Nitelikler	%10	**141**	**19736**	**88**
	PSO	%32.2	395	156375	362
	HŞO	%10.8	166	27451	122
	GKO	%13.2	188	35154	147
	YO	%32.2	395	156375	362
	GO	%10.6	149	22256	118
	YAO	%28.9	356	126449	323
Karar Ormanı	Tüm Nitelikler	%15.3	213	45478	157
	PSO	%25.2	327	106611	281
	HŞO	%19.6	239	56978	196
	GKO	%19.5	260	67752	206
	YO	%25.2	327	106611	281
	GO	**%8.5**	143	20549	100
	YAO	%11.1	156	24299	110



Şekil 3. En iyi modellerin tahmin sonuçları ile gerçek sipariş değerlerinin karşılaştırılması (a) A ürünü (b) B ürünü (c) C ürünü (d) D ürünü (Comparison of prediction results of the best models and actual order values (a) Product A (b) Product B (c) Product C (d) Product D)

5. Sonuçlar (Conclusions)

Üretim planlarındaki gecikmelerin önlenmesi için hammaddenin ve yarı mamulün belirlenen miktarda bulundurulması özellikle çelik sektöründe planların gerçekleşmesinde önemli bir etkiye sahiptir. Bu doğrultuda, tedarik zincirinde sürekliliğin sağlanması ve müşteriye teslim tarihlerinin gerçeklenmesinde talep tahmin modellerinin önemi göz ardı edilemez. Tedarik zinciri yönetiminin önemli bir süreci olan talep tahmini üzerine gerçekleştirilen çalışmalarda ortak amaç yüksek doğruluğa sahip bir tahmin modelinin geliştirilerek işletmelere fayda sağlamaktır. Bu çalışmada, çelik sektöründeki ürünlerin satış miktarları için yüksek doğrulukta tahmin değerleri elde edebilmek amacıyla, optimizasyon yöntemlerine dayalı nitelik seçimi yöntemlerinin makine öğrenmesi algoritmaları ile hibrit kullanımını içeren talep tahmin modelleri geliştirilmiştir. Nitelik seçimi yöntemleri olarak Parçacık Sürüsü Optimizasyonu, Harris Şahinleri Optimizasyonu, Gri Kurt Optimizasyonu, Yusufçuk Optimizasyonu, Genetik Optimizasyonu ve Yerçekimi Arama Optimizasyonu olmak üzere 6 optimizasyon algoritması kullanılmıştır. Modellerin geliştirilmesinde yararlanılan makine öğrenmesi algoritmaları ise Yapay Sinir Ağları, Karar Ormanı ve Lineer Regresyon algoritmaları olmuştur. Modeller çelik sektöründe faaliyet gösteren bir firmanın 4 ürününe ait gerçek verileri üzerinde eğitilerek test edilmiştir. Geliştirilen modeller MAPE, MAE, RMSE ve MSE performans ölçütlerine göre değerlendirilmiştir. Algoritmalar ile geliştirilen modellerin ortalama doğruluk değerleri incelendiğinde, yapay sinir ağlarıyla geliştirilen modellerin ortalama %91.9, karar ormanı yöntemi ile geliştirilen modellerin ortalama %91.3, lineer regresyon yöntemiyle geliştirilen modellerin ortalama %87.3 doğruluk değerine ulaştığı gözlemlenmiştir. Ancak, elde edilen sonuçlara göre, her bir ürün için farklı nitelik kombinasyonlarının ve farklı makine öğrenmesi yöntemlerinin üstün performans gösterdiği sonucuna ulaşılmıştır. Sonuç olarak, ürünler için en iyi tahmin performansı gösteren modeller %90.3-%94.7 arasında doğruluk değerleri sağlamıştır. Daha yüksek tahmin doğruluğu elde etmek amacıyla, gelecek çalışmalarda farklı metasezgisel nitelik seçimi yöntemleri ile derin öğrenmeye dayalı regresyon algoritmalarının uygulanması planlanmaktadır.

Kaynaklar **(References)**

Acı, M. ve Doğansoy, G. A. 2022. Makine öğrenmesi ve derin öğrenme yöntemleri kullanılarak e-perakende sektörüne yönelik talep tahmini. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 37(3), 1325-1340.

Aydın, M. R. ve Yazıcıoğlu, O. 2019. Yapay Sinir Ağları ile Talep Tahmini: Perakende Sektöründe Bir Uygulama. İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, 18(35), 43-55.

Bansal, P., Vanjani, A., Mehta, A., Kavitha, J. C. ve Kumar, S. 2022. Improving the classification accuracy of melanoma detection by performing feature selection using binary Harris hawks optimization algorithm. Soft Computing, 26(17), 8163-8181.

Catal, C., Kaan, E. C. E., Arslan, B. ve Akbulut, A. 2019. Benchmarking of regression algorithms and time series analysis techniques for sales forecasting. Balkan Journal of Electrical and Computer Engineering, 7(1), 20-26.

Chidroop, I. ve Moharir, M. 2020. Predicting the Propensity of Order Cancellation in the Ecommerce Domain. International Journal of Research in Engineering, Science and Management, 3(6). s. 658-664.

Chicco, D., Warrens, M. J., ve Jurman, G. 2021. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Computer Science, 7.

Diren, D. D., Boran, S., ve Cil, I. 2020. Integration of machine learning techniques and control charts in multivariate processes. Scientia Iranica, 27(6), 3233- 3241.

El Filali, A., El Filali, S. ve Jadli, A. 2022. Application of Deep Learning in the Supply Chain Management: A comparison of forecasting demand for electrical products using different ANN methods. In 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET) (pp. 1-7).

Elgamal, Z. M., Yasin, N. B. M., Tubishat, M., Alswaitti, M., ve Mirjalili, S. 2020. An improved harris hawks optimization algorithm with simulated annealing for feature selection in the medical field. IEEE access, 8, 186638-186652.

Fanoodi, B., Malmir, B. ve Jahantigh, F. F. 2019. Reducing demand uncertainty in the platelet supply chain through artificial neural networks and ARIMA models. Computers in biology and medicine, 113, 103415.

Feizabadi, J. 2022. Machine learning demand forecasting and supply chain performance. International Journal of Logistics Research and Applications, 25(2), 119-142.

Gökler, S. H. 2020. Kan Bankalarında Talep Tahmini ve Stokastik Stok Yönetimi. Doktora Tezi, Sakarya Üniversitesi.

Güven, İ. 2020. Perakende Hazır Giyim Endüstrisinde Yapay Zeka Yöntemleri ile Talep Tahmini. Doktora Tezi, Karabük Üniversitesi.

Han, G., Sönmez, E. F., Avcı, S. ve Aladağ, Z. 2022. Uygun Normalizasyon Tekniği ve Yapay Sinir Ağları Analizi ile Otomobil Satış Tahminlemesi. İşletme Ekonomi ve Yönetim Araştırmaları Dergisi, 5(1), 19-45.

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. ve Chen, H. 2019. Harris hawks optimization: Algorithm and applications. Future generation computer systems, 97, 849-872.

Huyen, C. 2022. Designing machine learning systems. O'Reilly Media.

Ismael, O. M., Qasim, O. S. ve Algamal, Z. Y. 2021. A new adaptive algorithm for v-support vector regression with feature selection using Harris hawks optimization algorithm. In Journal of Physics: Conference Series (Vol. 1897, No. 1, p. 012057). IOP Publishing.

İmece, S. ve Beyca, Ö. F. 2022. Demand Forecasting with Integration of Time Series and Regression Models in Pharmaceutical Industry. International Journal of Advances in Engineering and Pure Sciences, 34(3), 415-425.

Kacar, İ. 2024. Makine Öğrenimi Kullanarak Bir Mekanik Jiroskobun Yalpalama Tahmininde Zaman Serisi Modeli. Journal of Intelligent Systems: Theory and Applications, 7(1), 14-26.

Kennedy, J. 2010. Particle swarm optimization. In: Encyclopedia of Machine Learning, 760–766.

Keung, K. L., Lee, C. K. ve Yiu, Y. H. 2021. A machine learning predictive model for shipment delay and demand forecasting for warehouses and sales data. In 2021 ieee international conference on industrial engineering and engineering management (ieem).1010-1014. IEEE.

Kochak, A. ve Sharma, S. 2015. Demand forecasting using neural network for supply chain management. International journal of mechanical engineering and robotics research, 4(1), 96-104.

Korkut, D. 2019. Yapay sinir ağları yöntemi ile talep tahmini ve ayakkabı sektörüne uygulaması. Yayımlanmamış Yüksek Lisans Tezi., Hacı Bayram Veli Üniversitesi.

KS, S. R. ve Murugan, S. 2017. Memory based hybrid dragonfly algorithm for numerical optimization problems. Expert Systems with Applications, 83, 63-78.

Kück, M. ve Freitag, M. 2021. Forecasting of customer demands for production planning by local k-nearest neighbor models. International Journal of Production Economics, 231, 107837.

Lazzeri, F. 2020. Machine learning for time series forecasting with Python. John Wiley & Sons.

Lee, H. L., V. Padmanabhan ve S. Whang. 1997. “Information Distortion in a Supply Chain: the Bullwhip Effect.” Management Science 43: 546–558.

LightGBM. 2023, LightGBM Regressor, Erişim Tarihi:20.12.2023. https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMRegressor.html

Lingireddy, S. ve Ormsbee, L. E. 2002. Hydraulic network calibration using genetic optimization. Civil Engineering and Environmental Systems, 19(1), 13-39.

Merkuryeva, G., Valberga, A. ve Smirnov, A. 2019. Demand forecasting in pharmaceutical supply chains: A case study. Procedia Computer Science, 149, 3-10.

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

Mohammed, M., El-Shafie, H. ve Munir, M. 2023. Development and Validation of Innovative Machine Learning Models for Predicting Date Palm Mite Infestation on Fruits. Agronomy, 13(2), 494.

Mohan, B. A., Harshavardhan, B., Karan, S., Shariff, M. J. ve Pranav, M. G. 2021. Demand forecasting and route optimization in supply chain industry using data Analytics. In 2021 Asian Conference on Innovation in Technology (ASIANCON). 1-7. IEEE.

Muraina, I. 2022. Ideal dataset splitting ratios in machine learning algorithms: general concerns for data scientists and data analysts. In 7th International Mardin Artuklu Scientific Research Conference (pp. 496-504).

Orzechowski, A., Lugosch, L., Shu, H., Yang, R., Li, W. ve Meyer, B. H. 2023. A data-driven framework for medium-term electric vehicle charging demand forecasting. Energy and AI, 14, 100267.

Özçelik, T. Ö., Kibar, A. ve Bal, M.E., 2021. Sosyal Medyadan Veri Çekme Örnekleri. Mühendislikte Yapay Zeka ve Uygulamaları 4, Ed. Gülseçen, S., İnal, M.M., Torkul, O., Uçar, M.K., Sakarya Üniversitesi Yayınları, 79-101.

Poli, R., Kennedy, J. ve Blackwell, T. 2007. Particle swarm optimization: An overview. Swarm intelligence, 1, 33-57.

Sauro, J. ve Lewis, J. R. 2016. Quantifying the user experience: Practical statistics for user research. Morgan Kaufmann.

Spiliotis, E. 2022. Decision trees for time-series forecasting. Foresight, 1, 30-44.

Xu, S. ve Wang, S. 2022. Tourism Demand Prediction Model Using Particle Swarm Algorithm and Neural Network in Big Data Environment. Journal of Environmental and Public Health, 2022.

Tan, C. W., Dempster, A., Bergmeir, C. ve Webb, G. I. 2022. MultiRocket: multiple pooling operators and transformations for fast and effective time series classification. Data Mining and Knowledge Discovery, 36(5), 1623-1646.

Tavukçu, A. S. ve Sennaroğlu, B. 2021. Applying Forecasting Methods to Reduce the Cost of Spare Parts Inventory in a Company. Endüstri Mühendisliği, 32(3), 396-413.

Thaher, T., ve Arman, N. 2020. Efficient multi-swarm binary harris hawks optimization as a feature selection approach for software fault prediction. In 2020 11th International conference on information and communication systems (ICICS). 249-254. IEEE.

Thawkar, S. 2022. Feature selection and classification in mammography using hybrid crow search algorithm with Harris hawks optimization. Biocybernetics and Biomedical Engineering, 42(4), 1094-1111.

Torun, Z. ve DESTE, M. 2021. Sağlık İşletmelerinde Malzeme Yönetiminde Uygun Talep Tahmin Yönteminin Belirlenmesine Yönelik Bir Uygulama. 19 Mayıs Sosyal Bilimler Dergisi, 2(3), 581-613.

Türk, E. ve Kiani, F. Yapay Sinir Ağları ile Talep Tahmini Yapma: Beyaz Eşya Üretim Planlama Örneği. İstanbul Sabahattin Zaim Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 1(1), 30-37.

Yadav, A. ve Deep, K. 2013. Constrained optimization using gravitational search algorithm. National Academy Science Letters, 36, 527-534.

Yani, L. P. E., ve Aamer, A. 2023. Demand forecasting accuracy in the pharmaceutical supply chain: a machine learning approach. International Journal of Pharmaceutical and Healthcare Marketing, 17(1), 1-23.

Yaşar, H., Çağıl, G., Torkul, O. ve Şişci, M. 2021. Cylinder pressure prediction of an HCCI engine using deep learning. Chinese Journal of Mechanical Engineering, 34, 1-8.

Zeng, D., Chen, L., Zhao, S., Ou, J., Yuan, H. ve Wu, T. 2022. An Optimized Grey Wolf Algorithm. In 2022 IEEE International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC). 200-205. IEEE.

Research Article
Journal of Intelligent Systems: Theory and Applications 7(2) (2024) 81-94
DOI: 10.38016/jista.1439504

Conventional Machine Learning and Ensemble Learning Techniques in Cardiovascular Disease Prediction and Analysis

Buse Yaren Kazangirler[1*], Emrah Özkaynak[2]

[1] Department of Computer Engineering, Karabük University, Karabük, Türkiye

[2] Department of Software Engineering, Karabük University, Karabük, Türkiye

tekinbuseyaren@gmail.com, eozkaynak@karabuk.edu.tr

Abstract
Cardiovascular diseases, which significantly affect the heart and blood vessels, are one of the leading causes of death worldwide. Early diagnosis and treatment of these diseases, which cause approximately 19.1 million deaths, are essential. Many problems, such as coronary artery disease, blood vessel disease, irregular heartbeat, heart muscle disease, heart valve problems, and congenital heart defects, are included in this disease definition. Today, researchers in the field of cardiovascular disease are using approaches based on diagnosis-oriented machine learning. In this study, feature extraction is performed for the detection of cardiovascular disease, and classification processes are performed with a Support Vector Machine, Naive Bayes, Decision Tree, K-Nearest Neighbor, Bagging Classifier, Random Forest, Gradient Boosting, Logistic Regression, AdaBoost, Linear Discriminant Analysis and Artificial Neural Networks methods. A total of 918 observations from Cleveland, Hungarian Institute of Cardiology, University Hospitals of Switzerland, and Zurich, VA Medical Center were included in the study. Principal Component Analysis, a dimensionality reduction method, was used to reduce the number of features in the dataset. In the experimental findings, feature increase with artificial variables was also performed and used in the classifiers in addition to feature reduction. Support Vector Machines, Decision Trees, Grid Search Cross Validation, and existing various Bagging and Boosting techniques have been used to improve algorithm performance in disease classification. Gaussian Naïve Bayes was the highest-performing algorithm among the compared methods, with 91.0% accuracy on a weighted average basis as a result of a 3.0% improvement.

Keywords: Ensemble learning, classification, conventional techniques, cardiovascular disease, hyperparameter optimization.

Kardiyovasküler Hastalık Tahmini ve Analizinde Geleneksel Makine Öğrenmesi ve Topluluk Öğrenme Teknikleri

Öz
Kalp ve kan damarlarını önemli ölçüde etkileyen kardiyovasküler hastalıklar, dünya çapında önde gelen ölüm nedenlerinden biridir. Yaklaşık 19,1 milyon kişinin ölümüne neden olan bu hastalıkların erken teşhis ve tedavisi büyük önem taşıyor. Koroner arter hastalığı, kan damarı hastalığı, düzensiz kalp atışı, kalp kası hastalığı, kalp kapağı sorunları ve doğumsal kalp kusurları gibi birçok sorun bu hastalık tanımına girmektedir. Günümüzde kardiyovasküler hastalık alanındaki araştırmacılar tanı odaklı makine öğrenmesine dayalı yaklaşımlar kullanmaktadır. Bu çalışmada kardiyovasküler hastalık tespiti için özellik çıkarma işlemi gerçekleştirilmiş ve Destek Vektör Makinesi, Naive Bayes, Karar Ağacı, K-En Yakın Komşu, Torbalı Sınıflandırıcı, Rastgele Orman, Gradyan Artırım, Lojistik Regresyon, AdaBoost, Doğrusal Diskriminant Analizi ve Yapay Sinir Ağları yöntemleri ile sınıflandırma işlemleri yapılmıştır. Cleveland, Macaristan Kardiyoloji Enstitüsü, İsviçre Üniversite Hastaneleri ve Zürih VA Tıp Merkezi'nden toplam 918 gözlem çalışmaya dahil edilmiştir. Veri kümesindeki özellik sayısını azaltmak için bir boyut azaltma yöntemi olan Temel Bileşen Analizi kullanılmıştır. Deneysel bulgularda, özellik azaltmanın yanı sıra yapay değişkenlerle özellik artırımı da gerçekleştirilmiş ve sınıflandırıcılarda kullanılmıştır. Hastalık sınıflandırmasında algoritma performansını artırmak için Destek Vektör Makineleri, Karar Ağaçları, Izgara Arama Çapraz Doğrulama, var olan çeşitli Torbalama ve Artırma teknikleri kullanılmıştır. Gauss Naïve Bayes, %3,0'lık bir iyileştirme sonucunda ağırlıklı ortalama bazında %91,0 doğrulukla karşılaştırılan yöntemler arasında en yüksek performans gösteren algoritma olmuştur.

Anahtar Kelimeler: Topluluk öğrenme, sınıflandırma, geleneksel yöntemler, kardiyovasküler hastalık, hiperparametre optimizasyonu.

[*] Corresponding Author.
E-mail: tekinbuseyaren@gmail.com

Recieved : 19 Feb 2024
Revision : 15 Jun 2024
Accepted : 30 Jul 2024

1. Introduction

In recent years, Machine Learning (ML) studies in many sectors have continued sustainably without slowing down. The studies with sub-branches of Artificial Intelligence (AI), such as ML, pattern recognition, data science, and Deep Learning (DL), are vital in medicine. During the period when ML systematics were not used in medicine and health sciences, physicians and healthcare professionals were developing a manual approach while preparing diagnosis and treatment planning for patients. Therefore, with ML gaining a critical place today, it is concluded that it helps first-level physicians in health sciences to identify better patients who require additional attention and provide personalized tasks for each individual (Malik et al., 2019; Veranyurt et al., 2020). In various kinds of research, ML reveals an automated system to perform the desired task by extracting data-dependent statistical patterns (Chollet, 2021). Thus, computerized solutions become essential to treatment monitoring and planning, helping specialists reduce the adverse effects of time loss, stress, and fatigue in daily practice (Tekin et al., 2022).

Cardiovascular systems in the body of individuals consist of heart and blood vessels. Many various problems can occur in the cardiovascular system. Endocarditis, rheumatic heart disease, and abnormalities in the conduction system are shown as a few of the types of cardiovascular disease. Cardiovascular diseases are the leading cause of mortality in individuals worldwide (Lopez et al., 2022; Vatansever et al., 2021). When the causes of cardiovascular diseases in individuals are analyzed, modifiable and non-modifiable, i.e., congenital risk factors, stand out. These risk factors include adverse factors such as physical inactivity, long work hours, and family history. Regarding risk factors, non-modifiable factors such as age, gender, hypertension, and diabetes have different effects (Gregg and Hedayati, 2018). Family history, early atherosclerotic disease, or a first-degree relative after 55 years of age in men and after 65 years of age in women is recognized as a risk factor. In addition, in terms of gender, another non-modifiable factor, male individuals are more likely to have the disease than female individuals (Lopez et al., 2022). However, cardiovascular diseases, which are caused by many different causes, can also lead to other diseases. For this reason, disease monitoring is vital for diagnosing and treating high-risk patients in the early stages of the disease (Akman and Civek, 2022).

Many academic studies on cardiovascular diseases have been put forward when similar studies are examined in recent years. As a result of the research, while there are academic studies on the disease's risk factors, analysis, and examination determinations, ML needs to be adequately addressed. In 2016, Bektaş et al. (Bektaş and Babur, 2016) conducted a similar study in the health field and analyzed the performance of ML algorithms through feature selection methods on microarray datasets and prominent genes in breast cancer.

In 2018, Cihan (Cihan, 2018) performed a classification model with Random Forest (RF), 86.13% accuracy rate was obtained on the Cleveland dataset and an 86.13% accuracy rate was obtained on the dataset consisting of 596 patient records obtained by combining the Hungarian and Cleveland datasets. Badem (Badem, 2019) brought a different dimension to AI studies in health in 2019 by detecting Parkinson's disease using ML algorithms in audio signals. In addition to the algorithms used in the study, additional analysis was performed with Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) dimensionality reduction techniques. Veranyurt et al. (Veranyurt et al., 2020) used a dataset of 390 patients with 15 attributes to classify different types of diseases. As a result of the study, he compared the classification success of RF and K-Nearest. Neighbor (KNN) algorithms and achieved the highest success result.

In 2020, Taşçı and Şamlı (Taşçı and Şamlı, 2020) performed disease classification with WEKA on a cardiovascular disease dataset. Considering the studies in the literature, the use of 9 different algorithms and 13 attributes in Taşcı and Şamlı's study represents a significant contribution. In addition, the high number of features and relatively low number of cases can sometimes be considered limitations in studies. Although the accuracy rate with ZeroR, a data mining algorithm in their study, was relatively low at 49.18%, the other algorithms mentioned were able to achieve much higher scores with an average performance of 70-87%.

In 2021, another study was carried out to predict cardiovascular disease with Genetic Algorithm (GA) and other different algorithms. In this study, Vatansever et al. (Vatansever et al., 2021) put forward a research paper to analyze the risk factors that cause the disease. The open-source cardiovascular dataset was selected for the dataset used, and feature selection was performed on 14 features in this dataset. In the experimental results, the difference in performance before and after selection is noticeable. A high success rate was obtained with GA to contribute to the literature.

In 2022, Çil and Güneş (Çil and Güneş, 2022) performed a classification of heart diseases using Support Vector Machine (SVM), RF, Artificial Neural Network (ANN), Naive Bayes, and KNN algorithms. Dimensional reduction techniques and feature extraction were performed in the study. The backward elimination method removed insignificant features from the dataset and classified them. During learning, Cihan (Cihan, 2018) used all 11 attributes in the disease dataset. This may lead to unnecessary learning and need improvement in achieving the targeted performance. In addition, it was also concluded that no dimensionality reduction technique was used. Çil and Güneş ü,(Çil and Güneş, 2022) when the classification results of the algorithms they used in their study were analyzed, it was seen that while the precision success rate was very high, other metrics that should be considered in terms of performance were shallow. Accuracy value is sometimes not sufficient for a model to be considered successful.

The dominant aspects of the proposed model are clearly visible when compared with the models used in other studies. For example, while the highest accuracy rate in the Bektaş and Babur (2016) study was 90.7%, the proposed model surpasses this study with an accuracy rate of 95.00%. Additionally, Veranyurt et al. (2020) study, a maximum accuracy rate of 92.3% was achieved with RF, KNN and AdaBoost models, while the 95.00% accuracy rate achieved by the proposed model with various algorithms is beyond this study. Vatansever et al. (Vatansever et al., 2021) study, while an accuracy rate of 93.44% was achieved with various models, even the lowest accuracy rate of the proposed model was 81.00% and showed higher performance in general.

The research projects carried out between 2016 and 2024, the data sets, the machine learning models, and the experimental results are shown in Table 1. The identification of cardiovascular diseases using diverse datasets and ML algorithms has been the subject of numerous studies. These results show that the proposed model works with a wide range of algorithms, using a mixed data set consisting of a combination of various data sets, allowing to obtain higher accuracy scores in the detection of cardiovascular diseases. This reveals that the overall performance and reliability of the model are superior compared to other studies.

The main contribution of this work is to supplement the many algorithms used in the literature for the classification of cardiovascular disease with different conventional ML, ensemble methods, and ANN. PCA achieves dimensionality reduction using the correlation relation for each attribute used in disease detection (Abdi and Williams, 2010). In addition to the performance results obtained in the test runs after the training of the models, an optimization technique, Grid Search Cross-validation (CV) (Liashchynskyi and Liashchynskyi, 2019), was used to determine the best parameters for improvement. Another contribution of the study is the use of Boosting methods, alternative powerful ensemble learning techniques, in addition to the classification algorithms, and specially built ANN models classifier. Unlike other studies, optimized performance results have been achieved with more than one preprocessing technique, which will contribute to the literature.

2. Material and Methods

2.1. Preparation of the Dataset

The data considered in this study combines different datasets that exist independently but have yet to be connected before. The difference from the datasets in the literature is that four other dataset producers use the same variables to replicate the data and store them in a publicly available data store.

Table 1. Detailed review of studies on the detection and classification of cardiovascular disease.

Year	Author	Dataset	Model	Results (Accuracy)
2016	Bektaş and Babur (Bektaş and Babur, 2016)	Breast cancer, Kent Ridge 2 dataset	K-Star, Perceptron ANN, LibSVM, RF,	80.4%, 81.4%, 84.5%, 90.7%
2018	Cihan (Cihan, 2018)	Cleveland, Hungary, Switzerland,VA Long Beach dataset	RF	86.1%
2019	Badem (Badem, 2019)	Parkinson's disease classification dataset	DT, NB, SVM, RF, KNN	79.2%, 79.6%, 86.9%, 87.6%, 91.8%,
2020	Veranyurt et al. (Veranyurt et al., 2020)	Vanderbilt University Dept. of Biostatistics Diabetes dataset	AdaBoost, RF, KNN	90.5%, 92.3%, 92.3%,
2020	Taşcı et al. (Taşçı and Şamlı, 2020)	Cardiovascular disease dataset	ZeroR, OneR, DT, RF, LR, SVM, NB, KNN, Perceptron	49.1%, 73.7%, 78.6%, 83.6%, 85.2%, 86.8%, 86.8%, 88.5%
2021	Vatansever et al. (Vatansever et al., 2021)	USA Cleveland heart dataset	KNN, DT, RF, NB, SVM, GA, LR,	81.9%, 81.9%, 83.6%, 83.6%, 85.2%, 93.4%, 90.1%
2022	Çil and Güneş (Çil and Güneş, 2022)	USA CDC heart dataset	KNN, DT, ANN, RF, SVM, NB, LR	86.2%, 87.2%, 87.2%, 89.2, 90.5%, 90.5%, 90.7%
2024	**Our proposed model**	**Mixed heart disease dataset (combination of four dataset)**	**GB, XGBoost, DT, LR, LDA, KNN, RF, SVM, AdaBoost, GNBC, ANN**	**81.0%, 82.0%, 83.0%, 84.0%, 85.0%, 86.5%, 87.0%, 88.0%, 88.0%, 90.0%, 91.0%, 95.0%**

The original dataset includes 303 observations from the Cleveland Clinic Foundation, 293 observations from the Hungarian Institute of Cardiology, 123 observations from the Swiss University Hospitals, and 199 from the Long Beach VA Medical Centre (Zein Elabedin Mohammed et al., 2020). As a result of analyzing the information provided by individuals with cardiovascular diseases, 11 attributes created in the dataset are given in Table 2. When the dataset is analyzed, modifiable and innate attributes are housed together. The attribute "Cardiovascular Disease" as the target class is a numeric variable that produces the result 0 or 1.

2.2. Exploratory Data Analysis

The main modifiable risk factors affecting coronary cardiovascular diseases are overweight, diabetes, tobacco use, blood pressure, and cholesterol (Çil and Güneş, 2022). Therefore, the 6th attribute in the table, "FastingBS" is directly related to diabetes. Thus, as control problems increase daily in diabetic patients, blood pressure and total cholesterol levels also increase (Kara and Çınar, 2011). Another attribute, "Cholesterol" is a blood lubricant that forms a circulation found in all body cells. It was observed that FastingBS and cholesterol-derived risk factors indirectly matched with criteria such as gender, low physical activity, and family history (Çil and Güneş, 2022).

Table 2. Descriptions of the attribute's cardiovascular disease dataset.

Feature	Feature Type	Details of attributes
Age	Numerical	[28, 32, 42, ..., 77]
Sex	Nominal	[M: Male, F: Female]
ChestPainType	Nominal	[TA, ATA, NAP, ASY]
RestingBP	Numerical	[0, 80, 100, ..., 200]
Cholesterol	Numerical	[0, 120, 180, ..., 603]
FastingBS	Numerical	[0: False, 1: True]
RestingECG	Nominal	[Normal, ST-T, LVH]
MaxHR	Numerical	[60, 74, 88, ..., 202]
ExerciseAngina	Nominal	[Y: Yes, N: No]
Oldpeak	Numerical	[-2.6, 0.04, ..., 6.2]
ST-Slope	Nominal	[Y: Yes, N: No]
HeartDisease	Numerical	[0: Disease, 1: Normal]

As seen in Table 2, the variables are nominal, i.e., categorical, and numerical, i.e., numerical. For example, for FastingBS, if the value is more excellent than 120 mg, it represents 1, i.e., true, and if the value is less than 120 mg, it means 0, i.e., false. The risk factor "Sex'" nominally represents male for M (Male) and female for F (Female). For another attribute, "ChestPainType", TA represents typical angina, ATA represents atypical angina, NAP represents non-anginal pain, and ASY represents asymptomatic angina. Angina is a feeling of chest pain caused by spasms and pain in coronary cardiovascular disease. It is concluded that the existing attributes for angina measurements for "ExerciseAngina" and "ChestPainType" should be given to the algorithms for learning purposes. For "RestingECG", electrocardiogram measuring wave abnormality (T-wave inversions and ST elevation or depression of 0.05 mV), LVH indicates possible or definite left ventricular hypertrophy. Heart rate adjustment of ST-segment depression during exercise, performed by calculating the "Oldpeak" index, offers measurement of upsloping ST segments that may improve sensitivity with preservation of specificity from improved classification of patients with heart rate adjustment.

For the target category of the study, "HeartDisease" attribute, the total observations include 508 normal and 410 patient observations. The fact that these observations are chosen to be close to each other in terms of classification means that the algorithms are not prone to bias. Looking at the existing correlations with the target class for the attributes in the cardiovascular disease dataset, the results in Table 2 are obtained. However, we also set up a second dataset with 410 normal and 410 patient classes to check whether there was a problem with the fully balanced dataset in the experimental results. In order to avoid confusion in the study, 2 different datasets are denoted as Balanced: B, Unbalanced: UB to avoid confusion. Dataset B represents 410 normal 410 patient, while dataset UB represents 508 normal 410 patient.

Table 3. Feature correlation measurements for class of cardiovascular disease in the UB dataset after preprocessing.

Feature	Feature Type	Correlation Result
ST-Slope-Up	Numerical	-0.622164
ChestPainType-ATA	Numerical	-0.401924
MaxHR	Numerical	-0.400421
Cholesterol	Numerical	-0.232741
ChestPainType-NAP	Numerical	-0.212964
RestingECG-Normal	Numerical	-0.091580
ChestPainType-TA	Numerical	-0.054790
RestingECG-ST	Numerical	0.102527
RestingBP	Numerical	0.107589
FastingBS	Numerical	0.267291
Age	Numerical	0.282039
Sex-M	Numerical	0.305445
Oldpeak	Numerical	0.403951
ExerciseAngina-Y	Numerical	0.494282
ST-Slope-Flat	Numerical	0.554134

Regression analysis is a statistical technique for accommodating a cause-and-effect relationship. It is used for prediction (no prediction beyond the data used in the analysis), while correlation is used to determine the degree of the relationship (Asuero et al., 2006). In this study, assuming the number and dependency of the features, it is concluded that multiple regression analysis should be performed.

2.3. Preprocessing of Data

Preprocessing steps for data cleaning during data analysis are considered one of the essential steps in data-dependent studies in the literature. The dataset examined in the study is a mixed data source consisting of nominal and numerical values with 11 attributes. While 80% of a total of 701 observations were reserved for training, 20% were determined to be used in the testing phase. The correlation coefficient r revealed negative and positive correlation relationships for the target class, provided there were non-normalized features in the first step (Mintemur, 2021). The Label Encoder technique was used to digitize the nominal data. The components were standardized by removing the mean and scaling with the Standard Scaler, the next preprocessing step (Imad et al., 2022). The Standard Scaler technique is used to standardize the features. The correlation measurements between the components in the formed cluster and the target variable were calculated. Table 2 presents the new correlation values obtained. In this study, outlier data analysis and identification, which is another preprocessing step, was performed.

$$\mathrm{IQR} = \mathrm{Q3} - \mathrm{Q1} \quad (1)$$

Q1 in Equation 1 is the first quartile of the data, 25% of the data lies between the minimum and Q1. Q3 is the third quarter of the data, meaning 75% of the data falls between the minimum and Q3. The outliers to be reduced after the calculated Q3 and Q1 values are obtained by applying the observations that are less than or equal to Q3+1.5*IQR for the upper limit and greater than or equal to Q1-1.5*IQR for the lower limit (Perez and Tah, 2020). While outlier data were in the observations, observations were 918, and with the removal of outliers, observations were 701. Figure 1 belongs to the correlation matrix between the features after removing outliers with the interquartile range technique. Negative measurements between values in the matrix indicate that it has the opposite relationship with the target variable.

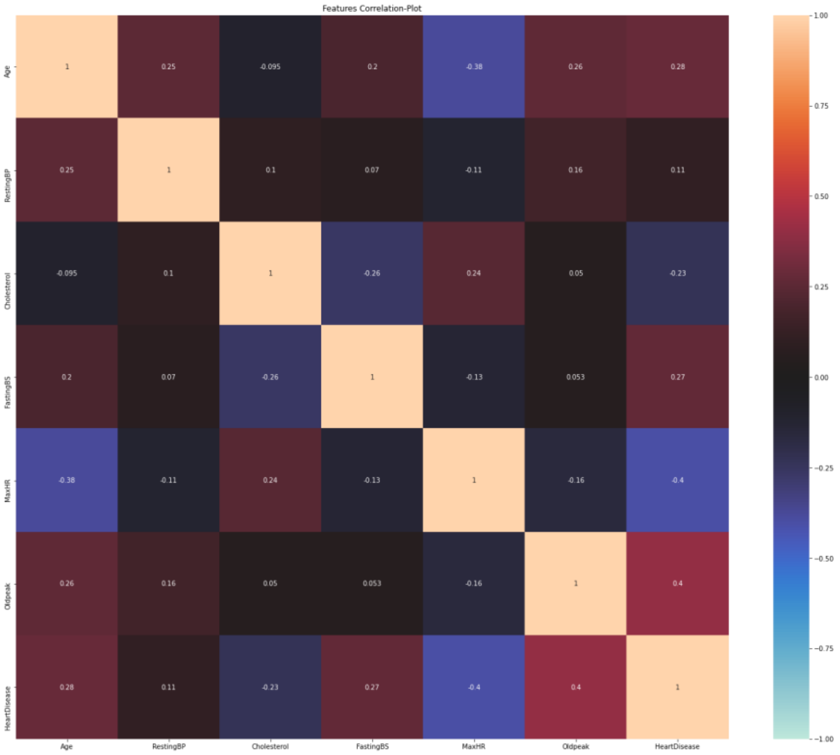


Figure 1. Identification of outliers in the data set with the interquartile range technique and the inter-feature correlation matrix in the UB dataset.

Accordingly, the diagonal is colored with the lightest color corresponding to +1, as there is ideal correspondence between the features. Measures with negative values in the matrix indicate they have the opposite relationship with the target variable. For example, a negative correlation exists between “Cholesterol” and the class “HeartDisease'”. After feature extraction according to the standardized observation data in the dataset, which was divided into training and test sets, the next preprocessing step was the dimensional reduction technique.

2.4. Dimensional Reduction Technique

The use of datasets with too many attributes for algorithms determined in ML projects leads to poor performance. The number of observations in the dataset should be high with the discovery of a certain amount of selection of features. Dimensional reduction techniques mean reducing unnecessary and redundant features in datasets. Reducing feature space with necessary feature selection and extraction ways is a proper statistical technique and a familiar method for discovering designs in high-dimensional data (Karamizadeh et al., 2013; Meng and Yang, 2012). PCA is one of the most famous techniques for reduction. To study a more down-dimensional space, the data is directed toward linear dimensionality reduction. The input data is centered. In the new variable space created by minimizing the cardiovascular dataset size, it is ensured that the most relevant features are in that space (Çil and Güneş, 2022). When Figure 2 is examined, it is concluded that maximum heart rate decreases with age and cardiovascular disease increases as maximum heart rate decreases. The "Age" and "MaxHR" attributes refer to the graph before and after pre-processing. As seen in the figure, correlation measurements were performed for all features. In this way, the connections of the features in the dataset with each other were also controlled formally. The data to be removed were determined by ranking the variance inflation factor and attribute values according to the principal component method. Instead of working with multiple original numerical features, linear combinations of them are obtained, paying attention to those that describe as many variations as possible from the original observations. Choosing linear combinations of predictors based on the maximum variance of the observations for the target variable "HeartDisease" was beneficial for prediction. Thus, the PCA transformation was carried out by providing dimension reduction. In PCA analysis, the error term is neglected in the calculation of the common factor variances of the features (Alkan, 2008).

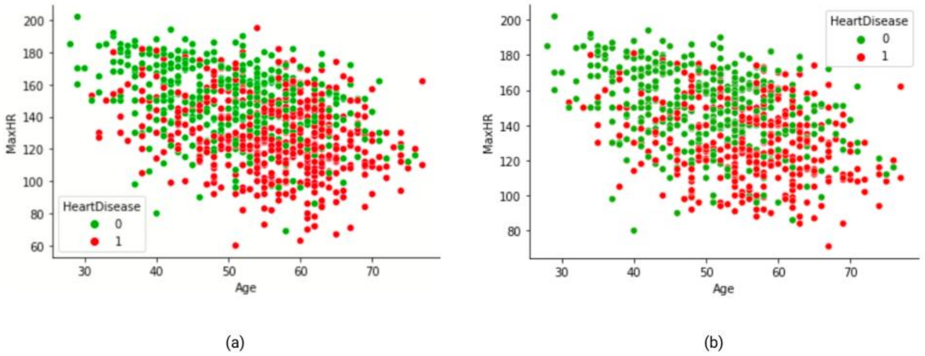


Figure 2. Correlation measures of age and maximum heart rate variables for cardiovascular disease. (a) correlation graph of age and maximum heart rate variables without pre-processing, (b) correlation graph of age and maximum heart rate variables as a result of pre-processing.

Figure 3 reveals the cumulative variance value by calculating the variance explained by the sum of the eigenvalues. In this step, 15 principal components were selected, and the variances explained by the components and the cumulated variance values were graphed. As can be seen, the variance of the first component is more meaningful than the other principal components. Therefore, the first 6 components may be sufficient to make sense of an average dataset. Components are calculated by capturing the variance in the data in the best way for dimension reduction with the PCA method. As seen in Figure 3, the plot shows the variance explained by each component against the number of components. According to these values, 6 principal components were selected as it is unnecessary to add additional components from the point where the curve flattens (Umargono et al., 2019). The curve breakpoint principle aims to select components that explain a large proportion of the total variance. Here, the point at which the plot bars and the curve become significantly flatter is designated as the break point. Therefore, component selection was performed where it did not provide a significant increase. Since the cumulative variance ratio reached sufficient saturation on this graph, 6 features were selected. The selection of these components is based on PCA analysis and the sum of the component loadings. The 6 most important features selected by PCA are Sex-F, Sex-M, RestingECG-ST, ST-Slope-Flat, RestingECG-Normal and RestingECG-LVH. Their values are 1.457506, 1.457506, 1.454983, 1.425109, 1.375829 and 1.322278 respectively.

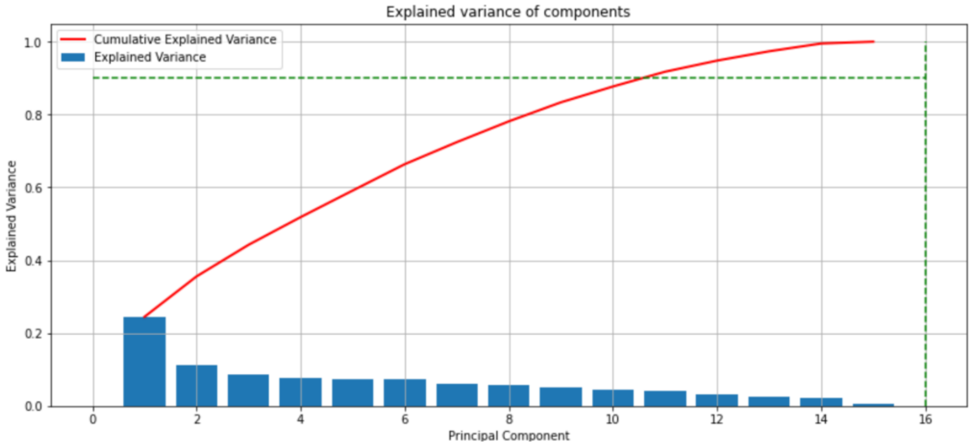


Figure 3. PCA analysis results total principal component count graph of the UB dataset.

2.5. Conventional Classification Techniques

After the preliminary preparation of the data, appropriate ML algorithms should be selected for the patterns to be found on the observations in the data sets. Classification is one of the supervised ML algorithms and is a frequently used task in studies with dependent features (Kaba and Kalkan, 2022).

Support Vector Classifier (SVC) is understanding from labeled training data to create estimations learning technique embedded in Structural Risk Minimization (SRM); it is among the well-known methods in machine learning (Cervantes et al., 2020; Moosaei et al., 2023). The support vectors are also recollection influential since they use a subset of the training topics. SVC's extraordinary generalization ability, optimal solution, and discriminating power have recently attracted attention. An infinite number of hyperplanes for linear separation of data are called optimal separation hyperplanes (Cristianini and Shawe-Taylor, 2000).

Naive Bayes Classifier (NBC) is a supervised learning algorithm that functions with the belief of "naive" dependent sovereignty between each couple with attributes and the class attribute. The training process of NBC is to predict the class preliminary probability based on the training set Zhang, 2004). GaussianNBC implements the Gaussian NBC algorithm for classification. The GaussianNBC classifier can be operated when the likelihoods of the features give the exact consequences (Pushpakumar et al., 2022). The classification problem in the study is to predict whether heart disease is present or absent.

Decision Trees are generally more rapid than artificial neural networks but do not have the suppleness to parameters Like SVCs, Decision Tree Classifiers (DTC) are practical techniques for appropriately challenging datasets (Singh et al., 2022). The aim is to make a technique that foresees a target variable. Additionally, the deeper the tree, the more complicated the rules and the more suitable the approach (Géron, 2022). The study handled this problem, and community learning techniques were used.

K-nearest Neighbor (KNN) techniques are an approach that is easy to implement but often runs quite slowly when the input dataset is huge. It is susceptible to extrinsic parameters. This classification algorithm, which has low efficiency due to lazy learning, is effective despite being a simple method (Guo et al., 2003). In this case, selecting the k parameter well is crucial to perform successfully. These are the resemblance measure between two data topics and the k's choice. The typical consequence of the foremost question is that various applications require various length sizes (Zhang, 2010; Zhang et al., 2017). Therefore, the choice of the k value merely uses the Euclidean length to compute the resemblance (Qin et al., 2007).

Logistic Regression (LR) is a particular point of approach with Binomial or Bernoulli distribution. The numerical result of the LR, which is the estimated likelihood, is used as a model. It is believed that target y_i accepts values in the set 0-1 for data point i. Once deployed, LR's prediction method predicts the probability of the positive class. LR is usually utilized to indicate the likelihood that a sample belongs to a specific class. If the estimated likelihood is greater than 50%, the model estimates that the sample belongs to that class (Géron, 2022).

Discriminant Analysis (DA) is one of the prevalent techniques for extracting the best features. It is developed as a problem to find an optimal value. It is also helpful but must be developed for nonlinear cases for more complicated ones (Kurita et al., 2009). Linear Discriminant Analysis (LDA) and Normal Discriminant Analysis (NDA) generalize Fisher's linear discriminant. Also, the algorithm supplies a Gaussian density to all types (Tharwat et al., 2017).

2.6. Ensemble Learning Techniques

Ensemble learning techniques are divided into two: bagging and boosting. In the bagging technique, new trees are created by repeatedly pulling samples from the dataset to be replaced. Then, a community emerges with the created trees. The boosting technique makes inferences from the ensemble by giving different weights to the dataset. One way to obtain various approaches is to utilize diverse techniques. Another technique is using the exact technique for each estimator. When sampling with replacement, this approach is called bagging. (Zhang et al., 2017). The Bagging Classifier (BC) is presented via Leo Breiman in 1994. This technique can use classification and regression methods. It is developed to enhance the strength and precision of ML approaches used. BC has received much attention for its simple implementation and increased accuracy. Therefore, it can be considered a “smoothing operation”, which is advantageous when improving the forecast performance of trees (Breiman, 2001; Géron, 2022).

A RF Classifier (RFC) is a group DTCs commonly trained by the bagging technique and generally with a maximum sample set. Rather than creating a GC and giving a DTC to it, it will likely utilize the RFC. The RFC algorithm provides an additional lacking pattern when growing trees; it explores the most helpful attribute. This source of randomness aims to reduce the variance of the forest predictor (Breiman, 2001). The prevailing opinion of most boosting strategies is to train estimators, each attempting to repair the earlier one. Many boosting methods are known, but the most famous are Adaptive Boosting (AdaBoostC) and Gradient Boosting (GBClassifier, GBC). The GBC algorithm makes a progressively forward extra model. At each stage, the n class number regression trees are provided for the adverse gradient of the loss function. The model adds estimators sequentially to an ensemble, each updating the previous one (Géron, 2022). Extreme Gradient Boosting Classifier (XGBC), the optimized version of the GBC, is highly enhanced and adaptable. Also, the XGBC is frequently considered crucial. This algorithm, which has a place in the literature as an ensemble learning algorithm, is considered excellent. A genetic algorithm has optimized the hyperparameter vector of the XGBC approach to enhance the forecast exactness and trustworthiness of the XGBoost model (Gu et al., 2022). An AdaBoostC is introduced and utilized to estimate the training set (Hastie et al., 2009). AdaBoostC has been demonstrated to be a thriving learning approach; it iteratively produces different vulnerable trainees and includes their results using the weighted plurality voting rule (Sun et al., 2016).

2.7. Artificial Neural Networks

DL is a branch of ML and, thus, pattern recognition and emanates from Artificial Neural Networks (ANNs) that affect the design of moving and processing data between neurons. For the sequential ANNs to be created, the model consisting of a single-layer stack connected sequentially is built. Since the first layer in the model will give an input vector, after the input size has been determined, the batch size should be chosen depending on the samples for the dataset. Then, a model suitable for the problem should be constructed. In this step, dense hidden layers with a certain number of neurons are added. It will use the Rectified Linear Unit (ReLU). The basic unit of deep neural networks are layers, which are data processing modules to be considered as filters for data. The data is taken as raw data to the layers for neural networks and reaches a level that will be more useful. Relevant layers have been added for the neural network to be built, and the selection of the activation function and loss function has been carried out (Chollet, 2021; Géron, 2022). The neural network in Figure 4 is obtained as a result of adding the relevant Dense layers by choosing a Binary Cross Entropy (BCE) loss function. This loss function performs the calculation of the cross-entropy loss between the real labels and the predicted labels.

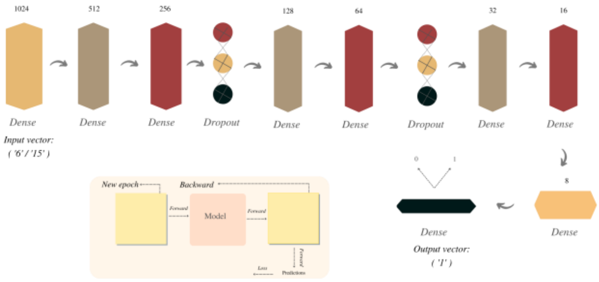


Figure 4. Neural network architecture suitable for cardiovascular disease prediction.

Since there is no categorical classification problem, it can be considered appropriate as a loss function since the cardiovascular disease result is 0 / 1. As activation functions, ReLU and Sigmoid functions were used respectively. Also, Mean Squared Error (MSE), which measures the mean of squares of errors is used. Thus, the mean of the sum of the squares of each difference between the predicted value and the true value was obtained. The network was trained for batch size: 2, optimizer: Adam, kernel initializer: Glorot uniform for a total of 500 epochs. While training the ANN, validation loss was continuously checked using Early Stopping techniques and training was terminated when the network stopped learning.

2.8. Hyperparameter Optimization with Grid Search Cross Validation and Randomized Search Cross Validation Techniques

In an ML study, hyperparameter optimization is the last step before experimental findings. Grid Search Cross Validation (GridSearchCV) is one of the various methods to discover a thriving and robust parameter for an algorithm. Grid search is a parameter-tuning approach to build and evaluate the selected model parameters (Ranjan et al., 2019). The n estimator parameters used in the approach were chosen at the level [10, 50, 100, 250, 500] (number of trees) to be transmitted to the classifier to be trained. In the evaluation procedure for the hyperparameter improvement part of the study, the model selection was provided by the RepeatedStratifiedKFold technique (Kramer, 2016). The parameter n is 3, and the number of folds is 10. For the values determined as the best parameters found in the AdaBoostC model as a result of GridSearchCV, the learning rate was 0.1, n estimators were 250, and the model result reached 87% accuracy. For the values determined as the best parameters found in the AdaBoostC model as a result of GridSearchCV, the learning rate was 0.1, n estimators were 250, the model result reached 87% accuracy. For the RF classifier, 64 candidates are selected for 10 folds in the same way and the algorithm is run. The maximize feature was 3, the minimum sample separation was 10, and the total number of trees was 200, and the best result was achieved with 90% accuracy for the classifier. Randomized Search Cross Validation (RandomizedSearchCV) is another method used for hyperparameter optimization. This method is similar to GridSearchCV, but requires less computational cost because it performs parameter searches over random samples rather than trying all possible combinations. The parameters of the RandomizedSearchCV model are optimized by a cross-validated search across many options, and unlike GridSearchCV, where all possible parameter values are tested, this method only tries a small subset of them from the selected distributions (Sharma et al., 2023). Accordingly, the method was applied for RFC and AdaboostC algorithms respectively. For the RFC algorithm, as in GridSearchCV, n estimators were trained to be 100, min samples split 20 and max features 3. As a result of testing the test set, an accuracy of 84.78% was obtained. In addition, AdaBoostC algorithm has set its best parameters according to RandomizedSearchCV technique with n estimators 100, learning rate 0.1. In this direction, the necessary training was performed and tested on the test set and the accuracy result was obtained as 84.42%.

2.9. Performance Evaluation Metrics

In ML studies, the confusion matrix reveals the connection between the class's ground truth classes and the model's estimated classes. Assessment of algorithm implementation is according to precision, recall, f1-score, and accuracy values in the equations in Equation 2, Equation 3, Equation 4, and Equation 5. Precision and recall metrics are often inversely proportional, as seen in Equation 2. F1-score is obtained from the harmonic average of the consequences in the equations to validate the optimization methods (Keser and Keskin, 2022; Tekin et al., 2022).

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 - score = 2x\frac{Precision*Recall}{Precision+Recall} \tag{3}$$

$$Accuracy = \frac{TP+TN}{TP + TN+FP+FN} \tag{4}$$

In the research, the Receiver Operator Characteristic (ROC) curve is often employed to demonstrate the efficiency of an algorithm. The ROC curve gives detailed knowledge about algorithm implementation and can be outlined as a single number area under the ROC Curve (AUC) (Meseci et al., 2022). AUC in Figure 5, revealed as an approach to calculate the performance, determines the accuracy of prediction in various techniques (Muschelli, 2020).

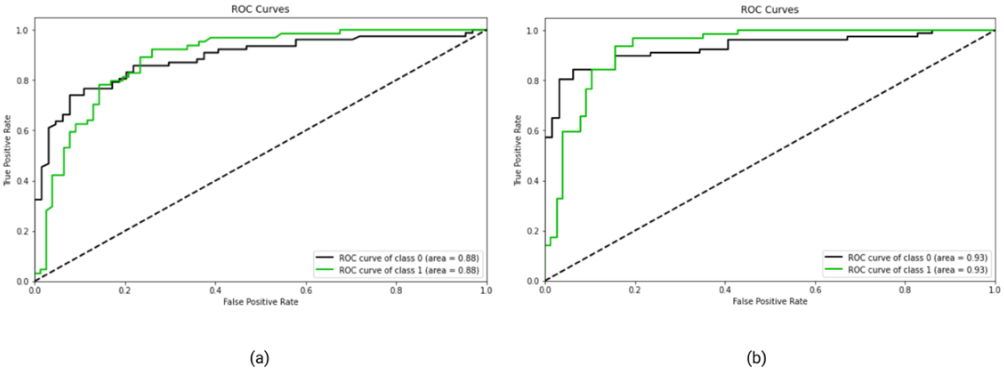


Figure 5. AUC graphs under the ROC curve in line with true positive and false positive rates for the worst and best classifier from the prediction scores in the UB dataset. (a) AUC-ROC graph for GBC, (b) AUC-ROC graph for the KNN classifier.

3. Experimental Results

Experimental results are the quantitative values obtained as a result of the studies performed during the evaluation of different types of ML models. This section includes the experimental findings before and after the pre-processing, as well as the performance results depending on the change of the attribute value. In addition to conventional classifiers in the literature such as SVC, NBC, DTC, KNN, LR, LDA, BaggingClassifier, RFC, GBC, AdaBoostC, etc. tree-based ensemble methods and ANN models such as were used. Thus, experimental findings that will contribute to the academic literature were obtained. As a result, while the weighted average accuracy was 71.0% for these two classes, the macro average accuracy was 70.0%. On top of that, when the classifier model was applied for the "linear, radial basis function" kernels with the GridSearchCV technique, the best score was obtained as 84.88% as a result of parameter selection. As a result of the cross-validation technique, the precision value for the "Normal" label was 85.0%, the recall value was 78.0% and the f1-score was 81.0%. As a result, the weighted average and macro accuracy for these two classes was 83.0%. Table 3 represents the experimental results obtained according to the features in the correlation matrix in Figure 1 after PCA analysis.

Table 4. Performance comparison table of preprocessing conventional and ensemble learning algorithms for the UB dataset.

Algorithm	Precision	Recall	F1-score	Accuracy
GBC	81.0%	80.9%	80.9%	81.0%
XGBC	82.0%	82.0%	82.0%	82.3%
DTC	84.0%	83.7%	83.7%	83.7%
RFC	**86.0%**	**86.5%**	**86.5%**	**86.5%**
KNN	**86.5%**	**86.0%**	**86.0%**	**87.0%**

In the next stage of the study, in addition to the previously mentioned features, a classification process was carried out with artificial indicators included in the dataset. In this case, instead 6 attributes, the categorical data in the data set was transformed into 15 artificial variables. Table 4 is based on the performance comparison of the classifier algorithms through 6 features. Table 5 shows the classification task results of 15 features included in the dataset as a result of the required pre-processing technique. In this case, performance improvement was observed for many classifiers and the algorithm with the best score was updated to GaussianNBC. Accuracy, ROC-AUC values are observed to show an improvement of 3.0%. In Table 6, the learning process is completed using BCE loss for 100 iterations. When the findings were analyzed, it was followed that the "Normal" class learned better, as expected. In Table 6, using the MAE loss metric, it is trained under the same conditions as the neural network used in BCE loss.

Table 5. Performance comparison table of conventional and ensemble learning algorithms by feature reduction for the UB dataset.

Algorithm	Accuracy	ROC	AUC
DTC	83.0%	82.0%	82.0%
LDA	84.0%	85.0%	85.0%
AdaBoostC	85.0%	85.0%	85.0%
KNN	85.0%	86.0%	86.0%
LR	85.0%	86.0%	86.0%
LinearSVC	85.0%	86.0%	86.0%
GaussianNBC	86.0%	86.0%	86.0%
RFC	**88.0%**	**87.0%**	**87.0%**
SVC	**88.0%**	**87.0%**	**87.0%**

When figure is carefully observed, it is concluded that the correlation connections increase with the variables "ChestPainType", "RestingECG" and "ST-Slope", which are not included in the 6-attribute classification problem. The features in the correlation matrix were used in classification and new values were added to the experimental findings.

Table 6. Performance comparison table of conventional and ensemble learning algorithms by feature increase for the UB dataset.

Algorithm	Accuracy	ROC	AUC
DTC	85.0%	85.0%	85.0%
AdaBoostC	87.0%	87.0%	87.0%
KNN	89.0%	89.0%	89.0%
SVC	89.0%	89.0%	89.0%
LDA	89.0%	89.0%	89.0%
LR	89.0%	89.0%	89.0%
LinearSVC	89.0%	89.0%	89.0%
RFC	90.0%	90.0%	90.0%
GaussianNBC	**91.0%**	**91.0%**	**91.0%**

Accordingly, evaluating an ANN algorithm is more suitable than many classifier approaches. When the results in the table are examined carefully, the precision value for the "HeartDisease" class is low, but the recall value is quite high. Therefore, it is concluded that there are too many false positive values. Contrary to Figure 1, a correlation matrix with more features is created and given in Figure 5.

Table 7. Performance comparison table of conventional and ensemble learning algorithms by feature reduction for the UB dataset.

Target	Precision	Recall	F1-score	Accuracy
HeartDisease-BCE	85.0%	84.0%	84.0%	87.0%
Normal-BCE	**90.0%**	**91.0%**	**91.0%**	**88.0%**
HeartDisease-MAE	82.0%	93.0%	87.0%	88.0%
Normal-MAE	**95.0%**	**85.0%**	**90.0%**	**89.0%**

In addition to the UB dataset, the algorithms used were also applied to the B dataset. Table 8 presents the performance comparison table of the conventional and ensemble learning algorithms for dataset B. According to the table, AdaBoostC, RF Classifier and SVC algorithms show the highest performance with a slight difference. In particular, RF Classifier and AdaBoostC algorithms outperform the other algorithms with 89.0% accuracy, ROC, and AUC values.

Table 8. Performance comparison table of conventional and ensemble learning algorithms for the B dataset.

Algorithm	Accuracy	ROC	AUC
DTC	82.0%	82.0%	81.5%
LDA	85.3%	85.0%	85.0%
LinearSVC	85.3%	85.4%	85.3%
LR	85.3%	85.4%	85.4%
XGBC	87.2%	87.3%	87.2%
KNN	87.2%	87.3%	87.2%
GaussianNBC	87.8%	87.7%	87.9%
GBC	88.4%	87.3%	87.3%
SVC	88.4%	88.5%	88.4%
AdaBoostC	89.0%	89.0%	89.0%
RFC	**89.0%**	**89.1%**	**89.0%**

These results show that ensemble methods and SVC algorithm perform better than other conventional algorithms and their performance improves.

In the feature increase process, 6 features obtained using PCA were transformed into artificial variables. This was done to better represent the data and improve the performance of the classification algorithms. The artificial variables were created using linear combinations of the original attributes, thus adding additional information to the dataset. As can be seen in Figure 5, new variables were selected for the main selected principal components taken from their internal categories. These attributes include interaction terms and higher-order polynomials of the original features. For example, the "Normal" and "ST" categories of the RestingECG attribute were taken as additional features, while the ATA, TA, and NAP attributes were added for ChestPainType, resulting in a total of 15 artificial variables.

Experimental findings show that feature reduction and increase techniques and hyperparameter optimization significantly improve the performance of the algorithms. The performance of the classifiers was significantly improved by using feature reduction and increase techniques. In particular, the best results were obtained when feature increase was applied by adding artificial variables. After these procedures, the Naive Bayes algorithm showed the highest performance with 91% accuracy. The best results were obtained with Naive Bayes, AdaBoostC and Random Forest algorithms. This study demonstrates the effectiveness of machine learning techniques in cardiovascular disease detection.

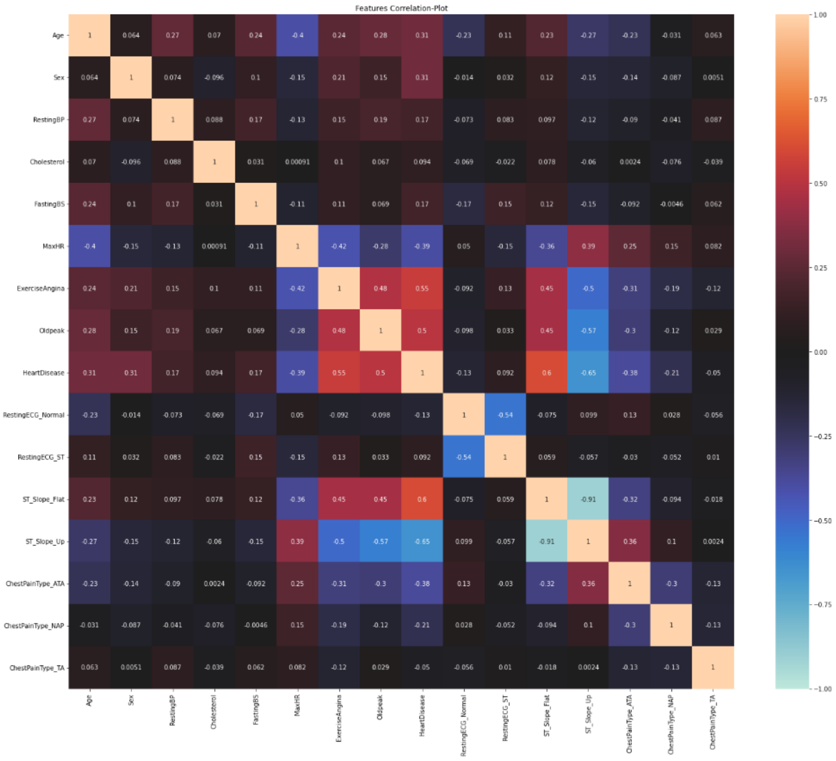


Figure 5. Correlation matrix of increasing features as a result of adding artificial variables for the UB dataset.

4. Conclusions

In this study, detection, and classification of cardiovascular disease with many algorithms was performed on a mixed dataset. Common algorithms in the literature were selected for classification and their success was increased according to the performance results obtained in similar studies. For this, both feature reduction and feature enhancement were applied by performing more than one pre-processing. In addition, statistically outlier data were cleaned with the IQR technique, and then improved by hyperparameter optimization with the GridSearchCV technique, a successful originality was demonstrated with a different approach compared to similar studies. With many ensemble learning techniques, algorithmically diverse results have been achieved. During the study, the correlation matrices were evaluated during each step, and the steps that gave the best performance were progressed in the process.

In the experimental findings section, all experiments carried out were meticulously supported by tables and figures. As a result of the study, the classifiers that gave the best results were GaussianNBC with 91.0%, RF Classifier with 88.0%, SVC with 88.0% and ANN model with 89.0% in the UB dataset. In addition, by providing hyperparameter optimization with the GridSearchCV technique, an improvement of approximately 3.0% was achieved in the results obtained in the experimental findings. Besides, RF Classifier was the algorithm that gave the highest score to the comparison table for dataset B. When the RF Classifier algorithm applied for the B dataset was compared with the result obtained for the UB dataset, it was concluded that there was a 1% performance increase.

This study successfully classified cardiovascular disease as a laborious and time-taking situation in the health field. Future studies and research aim to obtain more successful performances by minimizing the current error margin for detecting health problems, which is a difficult task.

5. Discussion

This study provides various machine learning and ensemble learning techniques are used for the detection and analysis of cardiovascular diseases. The results obtained are significant when compared to existing work in the literature. In this section, we will discuss the place and contributions of our work in the literature from a broad perspective. Research on the detection and analysis of cardiovascular diseases has made significant progress in recent years with the use of machine learning techniques. In their study, Bektaş and Babur (Bektaş and Babur, 2016) evaluated the performance of various machine learning algorithms for breast cancer diagnosis and obtained the highest accuracy rate of 90.7% with the RFC algorithm. Cihan (Cihan, 2018) demonstrated the effectiveness of the RFC algorithm with an accuracy rate of 86.1% using Cleveland and Hungary datasets.

In contrast to these studies, in our study, different reduction and augmentation techniques were applied for the features in the dataset for the detection of cardiovascular diseases and more algorithms were used. Feature reduction and enhancement techniques are frequently used to improve the performance of machine learning models. In our study, feature reduction was performed using PCA and then feature increase was applied by adding artificial variables. In particular, the Naive Bayes algorithm showed the highest performance with an accuracy of 91.0%. This result shows that the Naive Bayes algorithm can be effectively used in such classification problems.

In our study, bagging and boosting techniques and various ensemble learning algorithms were used. RFC and AdaBoostC algorithms are frequently used in the literature and have shown high performance (Breiman, 2001; Hastie et al., 2009). In this study, the RFC algorithm showed high performance with an accuracy of 88.0%. This result is consistent with the findings in the literature and confirms that the RFC algorithm is an effective method for cardiovascular disease detection. Moreover, ANN and deep learning techniques have achieved significant success in the medical field in recent years. In our study, the ANN model showed a high performance with an accuracy of 89.0%. This result shows that deep learning techniques are a powerful tool for the detection of cardiovascular diseases.

One of the most important contributions of this study is the comprehensive evaluation of the effectiveness of different machine learning and ensemble learning techniques in cardiovascular disease detection. In particular, the high accuracy rates achieved using feature augmentation with artificial variables and hyperparameter optimization are an important contribution to the literature. Our recommendation for future work is to improve the generalizability of the models using larger and more diverse datasets and to test different attribute reduction and augmentation techniques. Furthermore, evaluating the performance of deep learning models on more complex and larger datasets may contribute to better results in the detection of cardiovascular diseases.

References

Abdi, H., Williams, L.J., 2010. Principal component analysis. Wiley Interdiscip. Rev. Comput. Stat. 2, 433–459.

Akman, M., Civek, S., 2022. Dünyada ve Türkiye'de kardiyovasküler hastalıkların sıklığı ve riskin değerlendirilmesi. J. Turk. Fam. Physician 13, 21–28.

Alkan, Ö., 2008. Temel bileşenler analizi ve bir uygulama örneği. Atatürk Üniversitesi Sos. Bilim. Enstitüsü İşletme Anabilimdalı Üksek Lisans Tezi Erzurum 125s.

Asuero, A.G., Sayago, A., González, A.G., 2006. The correlation coefficient: An overview. Crit. Rev. Anal. Chem. 36, 41–59.

Badem, H., 2019. Parkinson Hastaliğinin Ses Sinyalleri Üzerinden Makine Öğrenmesi Teknikleri ile Tanimlanmasi. Niğde Ömer Halisdemir Üniversitesi Mühendis. Bilim. Derg. 8, 630–637.

Bektaş, B., Babur, S., 2016. Makine Öğrenmesi Teknikleri Kullanılarak Meme Kanseri Teşhisinin Performans Değerlendirmesi.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32.

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., Lopez, A., 2020. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing 408, 189–215.

Chollet, F., 2021. Deep learning with Python. Simon and Schuster.

Cihan, Ş., 2018. Koroner arter hastalığı riskinin makine öğrenmesi ile analiz edilmesi (PhD Thesis). Yüksek Lisans Tezi. Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü, Kırıkkale.

Çil, E., Güneş, A., 2022. Makine öğrenmesi algoritmalarıyla kalp hastalıklarının tespit edilmesine yönelik performans analizi. İstanbul Aydin Üniversitesi Dergisi Anadolu Bil Meslek Yüksekokulu.

Cristianini, N., Shawe-Taylor, J., 2000. An introduction to support vector machines and other kernel-based learning methods. Cambridge university press.

Géron, A., 2022. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc.

Gregg, L.P., Hedayati, S.S., 2018. Management of traditional cardiovascular risk factors in CKD: what are the data? Am. J. Kidney Dis. 72, 728–744.

Gu, Z., Cao, M., Wang, C., Yu, N., Qing, H., 2022. Research on Mining Maximum Subsidence Prediction Based on Genetic Algorithm Combined with XGBoost Model. Sustainability 14, 10421.

Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K., 2003. KNN model-based approach in classification. In: On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings. Springer, pp. 986–996.

Hastie, T., Rosset, S., Zhu, J., Zou, H., 2009. Multi-class adaboost. Stat. Interface 2, 349–360.

Imad, M., Abul Hassan, M., Hussain Bangash, S., Naimullah, 2022. A Comparative Analysis of Intrusion Detection in IoT Network Using Machine Learning. In: Big Data Analytics and Computational Intelligence for Cybersecurity. Springer, pp. 149–163.

Kaba, G., Kalkan, S.B., 2022. Kardiyovasküler Hastalık Tahmininde Makine Öğrenmesi Sınıflandırma Algoritmalarının Karşılaştırılması. İstanbul Ticaret Üniversitesi Fen Bilim. Derg. 21, 183–193.

Kara, K., Çınar, S., 2011. Diyabet bakım profili ile metabolik kontrol değişkenleri arasındaki ilişki. Kafkas J Med Sci 1, 57–63.

Karamizadeh, S., Abdullah, S.M., Manaf, A.A., Zamani, M., Hooman, A., 2013. An overview of principal component analysis. J. Signal Inf. Process. 4, 173.

Keser, S.B., Keskin, K., 2022. Ağırlıklı Oy Tabanlı Topluluk Sınıflandırma Algoritması ile Göğüs Kanseri Teşhisi. Mühendis. Bilim. Ve Araştırmaları Derg. 4, 112–120.

Kramer, O., 2016. Scikit-Learn. In: Kramer, O. (Ed.), Machine Learning for Evolution Strategies, Studies in Big Data. Springer International Publishing, Cham, pp. 45–53.

Kurita, T., Watanabe, K., Otsu, N., 2009. Logistic discriminant analysis. IEEE International Conference on Systems, Man and Cybernetics. Presented at the 2009 IEEE International Conference on Systems, Man and Cybernetics - SMC, IEEE, San Antonio, TX, USA, pp. 2167–2172.

Li, L., Zhou, Z., Bai, N., Wang, T., Xue, K.-H., Sun, H., He, Q., Cheng, W., Miao, X., 2022. Naive Bayes classifier based on memristor nonlinear conductance. Microelectron. J. 129, 105574.

Liashchynskyi, Petro, Liashchynskyi, Pavlo, 2019. Grid search, random search, genetic algorithm: a big comparison for NAS. ArXiv Prepr. ArXiv191206059.

Lopez, E.O., Ballard, B.D., Jan, A., 2022. Cardiovascular disease. In: StatPearls [Internet]. StatPearls Publishing.

Malik, P., Pathania, M., Rathaur, V.K., 2019. Overview of artificial intelligence in medicine. J. Fam. Med. Prim. Care 8, 2328.

Meng, J., Yang, Y., 2012. Symmetrical two-dimensional PCA with image measures in face recognition. Int. J. Adv. Robot. Syst. 9, 238.

Meseci, E., Ozkaynak, E., Dilmac, M., Ozdemir, D., 2022. PDC Dünya Dart Şampiyonası Karmaşık Ağlarında Komşuluk Tabanlı Bağlantı Tahmini. 5th Int. Conf. Data Sci. Appl. ICONDATA'22.

Mintemur, Ö., 2021. Doğrusal regresyonla vücut yağ tahmininde korelasyon türlerinin etkisi. EurasianSciEnTech 2021.

Moosaei, H., Ganaie, M.A., Hladík, M., Tanveer, M., 2023. Inverse free reduced universum twin support vector machine for imbalanced data classification. Neural Netw. 157, 125–135.

Muschelli, J., 2020. ROC and AUC with a Binary Predictor: a Potentially Misleading Metric. J. Classif. 37, 696–708.

Perez, H., Tah, J.H., 2020. Improving the accuracy of convolutional neural networks by identifying and removing outlier images in datasets using t-SNE. Mathematics 8, 662.

Platt, J., 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Adv. Large Margin Classif. 10, 61–74.

Pushpakumar, R., Prabu, R., Priscilla, M., Renisha, P.S., Prabu, R.T., Muthuraman, U., 2022. A Novel Approach to Identify Dynamic Deficiency in Cell using Gaussian NB Classifier. In: 2022 7th International Conference on Communication and Electronics Systems (ICCES). IEEE, pp. 31–37.

Qin, Y., Zhang, S., Zhu, X., Zhang, J., Zhang, C., 2007. Semi-parametric optimization for missing data imputation. Appl. Intell. 27, 79–88.

Ranjan, G.S.K., Verma, A.K., Radhika, S., 2019. K-nearest neighbors and grid search cv based real time fault monitoring system for industries. In: 2019 IEEE 5th International Conference for Convergence in Technology (I2CT). IEEE, pp. 1–5.

Sharma, N., Malviya, L., Jadhav, A., Lalwani, P., 2023. A hybrid deep neural net learning model for predicting Coronary Heart Disease using Randomized Search Cross-Validation Optimization. Decis. Anal. J. 9, 100331.

Singh, N., Jena, S., Panigrahi, C.K., 2022. A novel application of Decision Tree classifier in solar irradiance prediction. Mater. Today Proc. 58, 316–323.

Sun, B., Chen, S., Wang, J., Chen, H., 2016. A robust multi-class AdaBoost algorithm for mislabeled noisy data. Knowl.-Based Syst. 102, 87–102.

Tekin, B.Y., Ozcan, C., Pekince, A., Yasa, Y., 2022. An enhanced tooth segmentation and numbering according to FDI notation in bitewing radiographs. Comput. Biol. Med. 146, 105547.

Tharwat, A., Gaber, T., Ibrahim, A., Hassanien, A.E., 2017. Linear discriminant analysis: A detailed tutorial. AI Commun. 30, 169–190.

Umargono, E., Suseno, J.E., S. K., V.G., 2019. K-Means Clustering Optimization using the Elbow Method and Early Centroid Determination Based-on Mean and Median: In: Proceedings of the International Conferences on Information System and Technology. Presented at the International Conferences on Information System and Technology, Scitepress-Science and Technology Publications, Yogyakarta, Indonesia, pp. 234–240.

Vatansever, B., Aydın, H., Çetinkaya, A., 2021. Genetik algoritma yaklaşımıyla Öznitelik seçimi kullanılarak makine Öğrenmesi algoritmaları ile kalp hastalığı tahmini. J. Sci. Technol. Eng. Res. 2, 67–80.

Veranyurt, Ü., Deveci, A., Esen, M.F., Veranyurt, O., 2020. Makine Öğrenmesi Teknikleriyle Hastalık Sınıflandırması: Random Forest, K-nearest Neighbour ve Adaboost Algoritmaları Uygulaması. Uluslar. Sağlık Önetimi Ve Strat. Araşt. Derg. 6, 275–286.

Zein Elabedin Mohammed, A., Osama Fathy Kayed, M., Samy Abd El-Samee, M., 2020. Heart rate recovery time after excercise stress test in diabetic patients with suspected coronary artery disease. Al-Azhar Med. J. 49, 1845–1852.

Zhang, H., 2004. The optimality of naive Bayes. Aa 1, 3.

Zhang, S., 2010. KNN-CF approach: Incorporating certainty factor to knn classification. IEEE Intell Inform. Bull 11, 24–33.

Zhang, S., Li, X., Zong, M., Zhu, X., Cheng, D., 2017. Learning k for knn classification. ACM Trans. Intell. Syst. Technol. TIST 8, 1–19.

Research Article
Journal of Intelligient Systems: 7(2) (2024) 95-101
DOI: 10.38016/jista.1365609

SOM Clustering of OECD Countries for COVID-19 Indicators and Related Socio-economic Indicators

Pakize Yiğit[1*]

[1] İstanbul Medipol University,Medical School, Department of Medical Statistics and Medical Informatics, İstanbul, Türkiye

pyigit@medipol.edu.tr

Abstract

The coronavirus disease is one of the most severe public health problems globally. Governments need policies to better cope with the disease, so policymakers analyze the country's indicators related to the pandemic to make proper decisions. The study aims to cluster OECD (Organisation for Economic Co-operation and Development) countries using COVID-19, health, socioeconomic, and environmental indicators. A self-organizing map (SOM) clustering method, an unsupervised artificial neural network (ANN) method and a hierarchical clustering method are used. The data comprises 38 OECD countries, and 16 different variables are selected. As a result, the countries are grouped into 3 clusters. Cluster 1 contains 33 countries, the USA is Cluster 2, and Cluster 3 has 4 countries, including Turkey. COVID-19 mortality is highly related to mortality from chronic respiratory diseases. In addition, environmental indicators show differences in clusters.

Keywords: COVID-19, OECD countries, Kohonen SOM, clustering

1. Introduction

The recent coronavirus disease (COVID-19) pandemic is one of the serious public health problems in the World, causing 6,905,763 deaths worldwide on August 10th, 2020 (Worldometer, 2023). The World Health Organization (WHO) officially declared it as a Public Health Emergency of International Concern (PHEIC) on 30 January 2020. After three years, on 5 May 2023, the WHO Emergency Committee on the pandemic accepted that the disease did not fit a PHEIC. However, they warned that the condition is not over, so they continue giving suggestions to countries on how to manage the disease at the current time (WHO, 2023a).

The impacts of the pandemic caused difficulties for countries, especially in the field of health and economic systems. However, there were huge differences between countries reporting COVID-19 cases and death statistics due to these systems. Examining the variations between the countries is crucial for controlling the disease and reducing its burden (Gohari et al., 2022; WHO, 2023b).

Several studies have investigated the variations between the countries in terms of COVID-19 and related indicators. The studies examine the differences using different features such as social inequality and disease prevalence (Cardoso et al., 2020; Islam et al., 2021; Kumru et al., 2022), age and gender differences(Calderón-Larrañaga et al., 2020; Gebhard et al., 2020), environmental factors (Rizvi et al., 2021), lifestyle habits (smoking prevalence, alcohol consumption) (Aydin and Yurdakul, 2020; Kumru et al., 2022; Rizvi et al., 2021), health expenditures (Khan et al., 2020; Micah et al., 2021), healthcare capacity (Khan et al., 2020) and so many different aspects.

Furthermore, researchers have mostly used cluster analysis to examine differences between countries according to COVID-19 variables. Hussein and Abdulazeez (Hussein and Abdulazeez, 2021) reviewed the clustering algorithms applied to COVID-19 pandemic data. They stressed that K-means is the most widely used algorithm in this field with high accuracy. The using algorithms to detect variability of the countries COVID 19 related factors are K-Means clustering (Abdullah et al., 2022; Aydin and Yurdakul, 2020; Carrillo-Larco and Castillo-Cara, 2020; Gohari et al., 2022; Imtyaz et al., 2020; Rizvi et al., 2021; Siddiqui et al., 2020), hierarchical clustering (Aydin and Yurdakul, 2020; Sadeghi et al., 2021; Zarikas et al., 2020), fuzzy clustering (Mahmoudi et al., 2020), Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering algorithm (Shuai et al., 2020), Kohonen-SOM clustering (Boluwade, 2020).

Clustering analysis is one of the data mining methods in Big Data methodologies that use data

[*] Corresponding Author.
E-mail: pyigit@medipol.edu.tr

Recieved : 23 Sep 2023
Revision : 19 Dec 2023
Accepted : 5 Apr 2024

segmentation. In COVID-19 literature, traditional clustering methods (K-Medoids and Hierarchical clustering) are commonly used (Hussein and Abdulazeez, 2021) . In addition, Arunachalam and Kumar (Arunachalam ve Kumar, 2018) find that ANN-based SOM clustering finds hidden structures in the data set better than hierarchical, K-Medoids, and fuzzy clustering techniques. Bloom (2004) also found that SOM is better than the hierarchical clustering method, overcoming its limitations and better dealing with missing data (Brida vd., 2012). ANN makes predictions by mathematically modeling the way the human brain thinks. They do not require any statistical assumptions. Additionally, they succeed at identifying nonlinear models, so they are highly recommended for their flexibility, robustness, and higher prediction accuracy abilities to solve real-life problems. The SOM clustering method also applies a two-stage clustering method using both ANN and traditional clustering sequentially, which is more robust than other methods. Countries' COVID-19 data do not show normal distribution, and the relationships between variables are nonlinear. Therefore, SOM clustering analysis is effective in examining the differences and similarities between countries' COVID-19 and related variables.

Therefore, the present study aims to cluster OECD countries using COVID-19 and related socioeconomic indicators using SOM clustering method. It uses a two-level approach based on using a SOM in sequence, followed by hierarchical clustering analysis.

The rest of the paper is organized as follows: Section 2 presents material and methods and SOM analysis, Section 3 introduces findings, and Section 4 presents the conclusion.

2. Material and Methods

2.1. Dataset

In this paper, the analysis are performed on a data set of 38 OECD countries and 16 features. The data are obtained from different sources, OECD stat, World Bank, and our World in Data. The variables and their sources are presented in Table 1. The variables selected for the analysis are set according to related literature on COVID-19, mentioned in the introduction. They are examined under three different headings: COVID-19 variables (cum confirmed deaths, cum confirmed cases, cum vaccinations, cum tests), socioeconomic variables (life expectancy, elderly population, share of GDP, current PPPs, out-of-pocket health expenditures, smoking pr, alcohol consumption), environmental factors (EPI, HLT), diseases indicators (deaths from chronic respiratory diseases, deaths from cardiovascular diseases, diabetes pr). The newest available data is used for all countries.

The Environmental Performance Index (EPI) is calculated by researchers from Yale and Colombia Universities (Wolf et al., 2022). It was calculated from 40 indicators with 11 categories and three headings: environmental health (air quality, waste management, water, and sanitation, heavy metals), climate (climate change mitigation), ecosystem validity (biodiversity and habitat, ecosystem services, fisheries, agriculture, acid rain, and water resources). It uses both EPI and environmental health (HLT) indicators.

2.2. Kohonen SOM Analysis

SOM, also called Kohonen SOM, is an unsupervised ANN algorithm and introduced by Kohonen (Kohonen, 1982). The background of SOM comes from functions of neurons like other ANN methods. SOM can learn from multi-dimensional data and transform them into low-dimensional (mainly two-dimensional) topological order, preserving the original relations. The topological ordering map easily visualizes the similarities between the units according to their distance.

Table 1. Study Indicators

Variable name	Description	Data Source
Cum confirmed deaths	Total confirmed cases due to COVID-19 per million people	COVID-19 Indicator Our World in Data(Our World in Data, 2023)
Cum confirmed cases	Total confirmed deaths due to COVID-19 per million people	
Cum vaccinations	Total vaccinations per hundred	
Cum tests	Total COVID-19 test thousand people	
Life expectancy	The average measure of how long a born baby lives	Socioeconomic Indicators OECD(OECD, 2023)
Elderly population	percentage of aged 65 and over in the population	
Share of gross domestic product	The ratio of total health expenditures in gross domestic product (GDP)	
Current PPPs	Current health expenditure per capita	
Out-of-pocket health expenditures	Share of out of out-of-pocket health expenditure per capita	
Smoking pr	Daily smokers, % of population aged 15+	
Alcohol consumption	yearly sales of alcohol in liters per person aged 15+	
HLT	Measure of Environmental Health (HLT)	Environmental Performance Indicators(Wolf et al., 2022)
EPI	Measure of Environmental Performance Index (EPI)	
Deaths from Chronic respiratory diseases	Chronic respiratory diseases death rates (Sex: Both - Age:	Disease Mortality Our World In Data(Our World in Data, 2023)

	Age-standardized-2019)	
Deaths from Cardiovascular diseases	Cardiovascular disease death rates (Sex: Both - Age: Age-standardized-2019)	
Diabetes pr	Diabetes prevalence (% of population ages 20 to 79)	Disease Prevalence World Bank(The World Bank, 2023)

Like other ANN methods, it has input layer neurons (input data) and output layer neurons (topological order: hexagonal or rectangular lattice). The output layer neurons are connected to every neuron in the input nodes with weight vectors. The SOM algorithm is summarized into five stages (Haykin, 2008):

1. **Initialization:** Set the starting weight vectors wj(0) to random values.
2. **Sampling:** Draw a sample x with a certain probability from the input space. The activation pattern applied to the lattice is represented by the vector x. The dimension of the vector x is m.
3. **Similarity matching:** Utilizing the minimum-distance criterion, determine the best-matching (winning) neuron i(x) at time-step n:

$$i(x) = \arg\min_{j} \left\| x(n) - w_j \right\|, \quad j = 1,2,\ldots,l \tag{1}$$

4. **Updating:** Using the update formula, modify the synaptic-weight vectors of all stimulated neurons:

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i(x)}(n)\big(x(n) - w_j(n)\big) \tag{2}$$

5. **Continuation:** Use step 2 until there are no changes in the feature map.

2.3. *Application of SOM*

The Kohonen package (Wehrens, 2018) in R is used to perform SOM clustering. Firstly, the data is normalized and transformed into matrix form. In normalization, z values were used. The shape of the topological order should be chosen: hexagonal or circular. According to Kohonen's suggestion, the hexagonal topological order was selected (Kohonen, 2013). The number of nodes is decided as 5 $\sqrt{n}$ rule (Bruwer et al., 2018; Huiyan et al., 2008). The iteration demonstrates the iterative process and how distances arrive at their smallest value. After several experiments, a 5x5 SOM grid (25 neurons) with 1000-time iterations is created with hexagonal topologies. The learning rate was between 0.05 and 0.01.

A two-level approach based on using in sequence a SOM is used, followed by hierarchical clustering analysis, proposed by Vesanto and Alhoniemi (Vesanto and Alhoniemi, 2000). In this approach, initially, the SOM method applies and has SOM codes, and then SOM codes are clustered by hierarchical or partitive clustering methods. It helps to obtain more robust clusters. In this study, 25 neuron SOM codes clustered by Euclidean distance and Ward's agglomerative linkage method. Silhouette Index (Rousseeuw, 1987) and the Davies–Bouldin Index (Davies, D and Bouldin, D, 1979) are used to determine cluster size. Optimal clusters are found as three (Figure 1).

Spearman Correlation analysis is also conducted to measure the relationship between COVID-19 indicators and socioeconomic, environmental, and disease factors.

3. Results

The correlation matrix is given in Table 2. There are high positive correlations between COVID-19 cases and smoking pr (0.46) and alcohol consumption (0.52). Moderate correlation is found between COVID-19 cases and EPI (0.348). Moderate and negative correlation is also found between COVID-19 cases and life expectancy (-0.34). It is found high positive correlations between COVID-19 deaths and cardiovascular mortality rates (0.58) and smoking pr (0.52). In addition, high and negative correlation exists between COVID-19 deaths and life expectancy (-0.71) and HLT (-0.63). There is moderate correlation between COVID-19 deaths and alcohol consumption (0.39). High positive correlations are found between COVID-19 vaccinations and life expectancy (0.533) and HLT (0.51). There are moderate correlations between COVID-19 vaccinations and share of GDP (0.41), per capita current prices (0.41), and a negative, moderate correlation with cardiovascular mortality (-0.492).

Table 2. Spearman correlations for study Indicators

	Cum confirmed deaths	Cum confirmed cases	Cum Vaccinations	Cum tests
Life expectancy	-.712**	-.337*	.533**	-0.008
Share of GDP	-0.169	0.021	.412*	0.141
Current PPPs	-.394*	0.12	.409*	0.259
Out of Pocket Health Exp.	0.23	-0.179	-0.112	-0.098
HLT	-.631**	-0.113	.511**	0.248
EPI	-0.09	.348*	0.222	.464**
Elderly Population	0.196	0.159	0.085	0.199
Smoking pr	.519**	.461**	-0.113	.366*
Alcohol Consumption	.388*	.518**	-0.107	.400*
Deaths from Chronic respiratory diseases	-0.056	-0.102	0.141	-0.19
Deaths from Cardiovascular diseases	.579**	0.279	-.492**	0.068
Diabetes pr	0.13	-0.314	-0.247	-.369*

*p<0.05; **p<0.01

SOM quality is visually measured with node counts, node quality (distance), and SOM neighbor distance plots (Arunachalam and Kumar, 2018). They can be seen in Figure 2. The counts plot visualizes the number of countries in each node. There were 1-4 countries in each node. Grey nodes show empty nodes. The quality plot displays the average distance between countries. SOM neighbour distance plot (U-matrix) represents the distance between each node and its neighbors. It gives the idea of determining cluster numbers. The red color means closer neurons with similar characteristics, and the straw yellow indicates neighboring neurons with different characteristics.

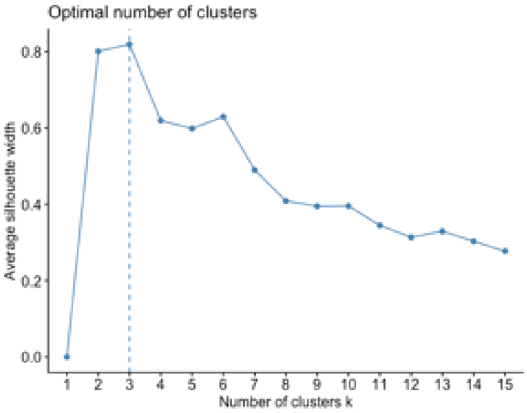


Figure 1. Silhouette Index

The membership of the 38 countries in the three clusters is provided in Table 3 and Figure 2.

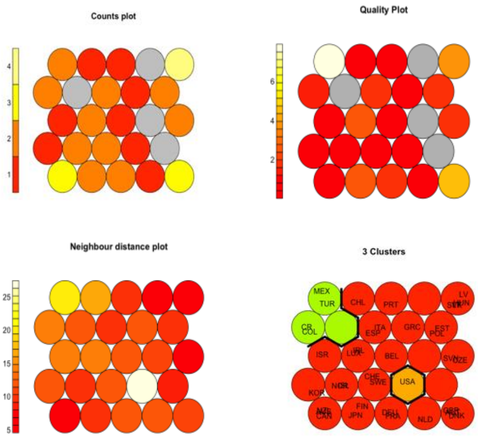


Figure 2. Counts Plot, Quality Plot, Neighbour distance plot and Cluster Plot

According to SOM visualization, it can be seen that The USA was the most different country across OECD countries. Secondly, Mexico and Turkey differ from other countries and are in the same nodes, meaning they have quite different characteristics from other countries, but highly similar each other.

The means of the study variables in the clusters are given in Table 4. Cluster 1 comprises 31 developed and two developing countries (Chile and Poland) (WEO Groups and Aggregates Information, 2023). Cluster 1 has the highest mean of cum confirmed cases (119,596), cum vaccinations (172), cum tests (2,966), life expectancy (80.7), the elderly population (19.23), HLT (77.06), EPI (60.38) and the lowest cum confirmed deaths (1,577), deaths from chronic respiratory diseases (20.72) and diabetes pr (6.25).

Table 3. Countries in the Clusters

Cluster-1			Cluster-2	Cluster-3
Australia	Greece	Latvia	United States	Colombia
Austria	Hungary	Israel		Costa Rica
Belgium	Iceland	Italy		Mexico
Canada	Lithuania	Japan		Turkey
Chile	Luxembourg	Portugal		
Czech Republic	Netherlands	Slovak Republic		
Denmark	New Zealand	Slovenia		
Estonia	Norway	Spain		
Finland	Poland	Sweden		
France	Ireland	Switzerland		
Germany	Korea	United Kingdom		

The USA, one of the developed countries, constitutes Cluster 2. The cluster-2 has the highest value of cum confirmed deaths (2,421), the share of GDP (17.36), current PPPs (12,196), and deaths from chronic respiratory diseases (37.72), and the lowest life expectancy (76.4), out-of-pocket health expenditures (10.70).

The cluster 3 comprises four developing countries: Colombia, Costa Rica, Mexico, and Turkey. They have the highest mean of out-of-pocket health expenditure (23.00), diabet pr (12.13) and the lowest cum confirmed cases (87,886), cum vaccinations (136.3), elderly population (8.85), share of GDP, current PPPs (1,506), smoking pr (15.00), alcohol consumption (3.43), HLT (48.60), EPI (40.13).

Smoking pr and alcohol consumption values of Cluster 1 and Cluster 2 are found to be very close to each other (23.70; 23.00 - 9.21; 9.50, respectively). Likewise, deaths from cardiovascular diseases also had very similar values for the three clusters (158.97; 157.01; 158.68, respectively). In addition, developed countries have the highest HLT and EPI, while developing countries have the lowest.

Table 4. Cluster mean of variables

Study Variable	Cluster1	Cluster2	Cluster3
Cum confirmed deaths	1,578	2,420	1,817
Cum confirmed cases	119,596	158,249	87,886
Cum vaccinations	171.82	157.20	136.30
Cum tests	2,966	2,155	628
Life expectancy	80.73	76.40	77.90
Elderly population	19.23	16.83	8.85
Share of gross domestic product	9.83	17.36	6.81
Current PPPs	4,877	12,196	1,506
Out-of-pocket health expenditures	18.00	10.70	23.00
Smoking pr	23.70	23.00	15.00
Alcohol consumption	9.21	9.50	3.43
HLT	77.06	76.80	48.60
EPI	60.38	51.10	40.13
Deaths from chronic respiratory diseases	20.72	37.72	34.27
Deaths from cardiovascular diseases	158.97	157.01	158.68
Diabetes pr	6.25	10.70	12.13

4. Conclusion

COVID-19 has affected many people globally, making it one of the most significant challenges to humankind recently. Since the beginning of the pandemic, various studies have been conducted to help policymakers make better decisions for countries. The study investigates and presents the topic using a robust clustering technique and using various indicators related the pandemic. Therefore, the study proposes to cluster OECD countries using COVID-19, health, socioeconomic, and environmental indicators.

This study uses the Kohonen SOM clustering method to cluster 38 OECD countries based on COVID-19 confirmed cases, deaths, vaccinations, tests, and health, socioeconomic, and environmental variables. The data set used in the study consists of 16 variables. The study conducted a two-level approach of clustering SOM: SOM and hierarchical clustering. Silhouette and the Davies–Bouldin Index methods were used to decide the optimal number of clusters, and the number of optimal cluster is three.

Cluster 1 has 33 countries, with 31 developed and two developing countries (Chile and Poland) showing the lowest mean of confirmed COVID-19 deaths and the highest confirmed cases, vaccinations, and tests. Cluster 2 consists of only USA. It distinguishes itself from other countries by having the highest number of COVID-19 deaths. Cluster 3 contains four developing countries: Colombia, Costa Rica, Mexico, and Turkey. It shows the lowest number of COVID-19 confirmed cases, vaccinations, and tests.

It is found that deaths from cardiovascular diseases are not distinctive in separating clusters because it is the leading cause of death for all countries in the World (WHO, 2020). It has almost similar risks for all countries. On the other hand, chronic respiratory disease mortality is strongly associated with COVID-19 indicators and confirmed by several studies (Kumru et al., 2022; Rizvi et al., 2021). The developed countries have higher EPI and HLT, lower COVID-19 mortality, confirming by other studies (Coccia, 2021; Rizvi et al., 2021). However, The USA has the highest COVID-19 mortality, GDP, and health expenditures. Studies show that the USA did not prevent COVID-19 cases surveillance and provide equal health services during the pandemic because of it substantial regional differences (Bergquist et al., 2020; Bollyky et al., 2023). In addition, Turkey is in the same cluster with three other developing countries. Developing countries have lower GDP, and health expenditures so their health system is weak to deal with the pandemic (Coccia, 2021). The COVID-19 statistics data is not also well documented in developing countries because of their health system (Levin et al., 2022).

The study differs from other literature to cluster countries. (1) it uses COVID-19 cases, deaths, tests and vaccinations, and related socioeconomic and environmental indicators. (2) It uses an ANN-based SOM clustering technique. Other studies have used different clustering methods and variables when investigating the hidden structures of COVID-19-related factors across countries.

There were some limitations in this study. First, the study examined only 38 OECD countries. Hence, the results may not fully reflect global trends in COVID-19 and related factors. Second, our data set did not contain all the variables related to the pandemic. Despite these limitations, the study uses an ANN-based SOM two-phased clustering approach with various related COVID-19 features. It has focused on OECD countries to have more quality comparatives especially for Turkey's situation. The hope is that the study helps policymakers make regulations about emergencies in our country and other countries and to plan new studies.

References

Abdullah, D., Susilo, S, Ahmar A.S., et al., 2022. The application of K-means clustering for province clustering in Indonesia of the risk of the COVID-19 pandemic based on COVID-19 data. *Quality and Quantity* 56(3). Springer Netherlands: 1283–1291. DOI: 10.1007/s11135-021-01176-w.

Arunachalam, D., Kumar, N., 2018. Benefit-based consumer segmentation and performance evaluation of clustering approaches: An evidence of data-driven decision-making. *Expert Systems with Applications* 111. Elsevier Ltd: 11–34. DOI: 10.1016/j.eswa.2018.03.007.

Aydin N and Yurdakul G., 2020. Assessing countries' performances against COVID-19 via WSIDEA and machine learning algorithms. *Applied Soft Computing Journal* 97. Elsevier B.V.: 106792. DOI: 10.1016/j.asoc.2020.106792.

Bergquist, S., Otten, T., Sarich, N., 2020. COVID-19 pandemic in the United States. *Health Policy and Technology* 9(4). Elsevier Ltd: 623–638. DOI: 10.1016/j.hlpt.2020.08.007.

Bollyky, T.J., Castro, E., Aravkin, A.Y., et al., 2023. Assessing COVID-19 pandemic policies and behaviours and their economic and educational trade-offs across US states from Jan 1, 2020, to July 31, 2022: an observational analysis. *The Lancet* 401(10385). The Authors. Published by Elsevier Ltd. This is an Open Access article under the CC BY 4.0 license: 1341–1360. DOI: 10.1016/S0140-6736(23)00461-0.

Boluwade, A., 2020. Regionalizing Partitioning Africa's Coronavirus (COVID-19) Fatalities Using Environmental Factors and Underlying Health Conditions for Social-economic Impacts. *2nd Novel Intelligent and Leading Emerging Sciences Conference, NILES 2020*: 439–443. DOI: 10.1109/NILES50944.2020.9257875.

Bruwer, J., Prayag, G., Disegna, M., 2018. Why wine tourists visit cellar doors: Segmenting motivation and destination image. *International Journal of Tourism Research* 20(3): 355–366. DOI: 10.1002/jtr.2187.

Calderón-Larrañaga, A., Dekhtyar, S., Vetrano, D.L., et al., 2020. COVID-19: risk accumulation among biologically and socially vulnerable older populations. *Ageing Research Reviews* 63(May). DOI: 10.1016/j.arr.2020.101149.

Cardoso, E.H.S., Silva, M.S., Da, Júnior, FEDAF, et al., 2020. Characterizing the Impact of Social Inequality on COVID-19 Propagation in Developing Countries. *IEEE Access* 8: 172563–172580. DOI: 10.1109/ACCESS.2020.3024910.

Carrillo-Larco, R.M., Castillo-Cara, M., 2020. Using country-level variables to classify countries according to the number of confirmed COVID-19 cases: An unsupervised machine learning approach. *Wellcome Open Research* 5: 56. DOI: 10.12688/wellcomeopenres.15819.1.

Coccia, M., 2021. High health expenditures and low exposure of population to air pollution as critical factors that can reduce fatality rate in COVID-19 pandemic crisis: a global analysis. *Environmental Research* 199(January). Elsevier Inc.: 111339. DOI: 10.1016/j.envres.2021.111339.

Davies, D.L., Bouldin, D.W., 1979. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2): 224–227.

Gebhard, C., Regitz-Zagrosek, V., Neuhauser, H.K., et al., 2020. Impact of sex and gender on COVID-19 outcomes in Europe. *Biology of Sex Differences* 11(1). Biology of Sex Differences: 1–13. DOI: 10.1186/s13293-020-00304-9.

Gohari, K., Kazemnejad, A., Sheidaei, A., et al., 2022. Clustering of countries according to the COVID-19 incidence and mortality rates. *BMC Public Health* 22(1). BioMed Central: 1–12. DOI: 10.1186/s12889-022-13086-z.

Haykin, S., 2008. *Neural Networks and Learning Machines*. DOI: 978-0131471399.

Huiyan, S.B., Gelfand, A.E., Chris, L., et al., 2008. Interpreting self-organizing maps through space–time data models. *The Annals of Applied Statistics* 2(4): 1194–1216. DOI: 10.1214/08-AOAS174.

Hussein, H.A., Abdulazeez, A.M., 2021. Covid-19 Pandemic Datasets Based on Machine Learning Clustering Algorithms: A Review. *Journal Of Archaeology Of Egypt/Egyptology* 18(4): 2672–2700. Available at: https://archives.palarch.nl/index.php/jae/article/download/6703/6488.

Imtyaz, A., Abid Haleem, Javaid, M., 2020. Analysing governmental response to the COVID-19 pandemic. *Journal of Oral Biology and Craniofacial Research* 10(4). Elsevier: 504–513. DOI: 10.1016/j.jobcr.2020.08.005.

Islam, N., Lacey, B., Shabnam, S., et al., 2021. Social inequality and the syndemic of chronic disease and COVID-19: County-level analysis in the USA. *Journal of Epidemiology and Community Health* 75(6): 496–500. DOI: 10.1136/jech-2020-215626.

Khan, J.R., Awan, N., Islam, M.M., et al., 2020. Healthcare Capacity, Health Expenditure, and Civil Society as Predictors of COVID-19 Case Fatalities: A Global Analysis. *Frontiers in Public Health* 8(July): 1–10. DOI: 10.3389/fpubh.2020.00347.

Kohonen, T., 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43(1): 59–69. DOI: 10.1007/BF00337288.

Kohonen, T., 2013. Essentials of the self-organizing map. *Neural Networks* 37. Elsevier Ltd: 52–65. DOI: 10.1016/j.neunet.2012.09.018.

Kumru, S., Yiğit, P., Hayran, O., 2022. Demography, inequalities and Global Health Security Index as correlates of COVID-19 morbidity and mortality. *International Journal of Health Planning and Management* 37(2): 944–962. DOI: 10.1002/hpm.3384.

Levin, A.T., Owusu-Boaitey, N., Pugh, S., et al., 2022. Assessing the burden of COVID-19 in developing countries: Systematic review, meta-Analysis and public policy implications. *BMJ Global Health* 7(5): 1–17. DOI: 10.1136/bmjgh-2022-008477.

Mahmoudi, M.R., Baleanu, D., Mansor, Z., et al., 2020. Fuzzy clustering method to compare the spread rate of Covid-19 in the high risks countries. *Chaos, Solitons and Fractals* 140. Elsevier Ltd: 1–9. DOI: 10.1016/j.chaos.2020.110230.

Micah, A.E., Cogswell, I.E., Cunningham, B., et al., 2021. Tracking development assistance for health and for COVID-19: a review of development assistance, government, out-of-pocket, and other private spending on health for 204 countries and territories, 1990–2050. *The Lancet* 398(10308): 1317–1343. DOI: 10.1016/S0140-6736(21)01258-7.

OECD, 2023. OECD Statistics. Available at: https://stats.oecd.org/ (accessed 11 July 2023).

Our World in Data (2023) Data, Coronavirus Pandemic (COVID-19) - Statistics and Research - Our World in. Available at: https://ourworldindata.org/explorers/coronavirus (accessed 8 July 2023).

Rizvi, S.A., Umair, M., Cheema, M.A., 2021. Clustering of countries for COVID-19 cases based on disease prevalence, health systems and environmental indicators. *Chaos, Solitons and Fractals* 151. Elsevier Ltd: 111240. DOI: 10.1016/j.chaos.2021.111240.

Rousseeuw, P.J., 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20(C): 53–65. DOI: 10.1016/0377-0427(87)90125-7.

Sadeghi, B., Cheung, R.C.Y., Hanbury, M., 2021. Using hierarchical clustering analysis to evaluate COVID-19 pandemic preparedness and performance in 180 countries in 2020. *BMJ Open* 11(11): 1–11. DOI: 10.1136/bmjopen-2021-049844.

Shuai, Y., Jiang, C., Su, X., et al., 2020. A Hybrid Clustering Model for Analyzing COVID-19 National Prevention and Control Strategy. *2020 IEEE 6th International Conference on Control Science and Systems*

Engineering, ICCSSE 2020: 68–71. DOI: 10.1109/ICCSSE50399.2020.9171941.

Siddiqui, M.K., Morales-Menendez, R., Gupta, P.K., et al., 2020. Correlation between temperature and COVID-19 (suspected, confirmed and death) cases based on machine learning analysis. *Journal of Pure and Applied Microbiology* 14(May): 1017–1024. DOI: 10.22207/JPAM.14.SPL1.40.

The World Bank, 2023. The World Bank Data.

Vesanto, J., Alhoniemi, E., 2000. Clustering of self-organizing map. *IEEE TRANSACTIONS ON NEURAL NETWORKS* 11(3): 586–600.

Wehrens, M.R., 2018. Package ‘ kohonen .’

WEO Groups and Aggregates Information, 2023. World Economic Outlook Database - Groups and Aggregates. Available at: https://www.imf.org/en/Publications/WEO/weo-database/2023/April/groups-and-aggregates (accessed 12 August 2023).

WHO, 2020. The top 10 causes of death. Available at: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death (accessed 28 December 2023).

WHO, 2023a. Coronavirus disease (COVID-19) pandemic. Available at: https://www.who.int/europe/emergencies/situations/covid-19 (accessed 10 August 2023).

WHO, 2023b. WHO Coronavirus (COVID-19) Dashboard.

Wolf, M.J., Emerson, J.W., Esty, D.C., et al., 2022. Environmental Performance Index. DOI: 10.1002/9781118445112.stat03789.

Worldometer, 2023. COVID - Coronavirus Statistics - Worldometer. Available at: https://www.worldometers.info/coronavirus/ (accessed 10 August 2023).

Zarikas, V., Poulopoulos, S.G., Gareiou, Z., et al., 2020. Clustering analysis of countries using the COVID-19 cases dataset. *Data in Brief* 31. Elsevier Inc.: 105787. DOI: 10.1016/j.dib.2020.105787.

Araştırma Makalesi
Zeki Sistemler Teori ve Uygulamaları Dergisi 7(2) (2024) 102-115
DOI: 10.38016/jista.1400047

Olay Kamerası ile Verimli Konuşma Sesi Tespiti için Zamansal Evrişimsel Ağlar

Arman Savran[1*]

[1*] Bilgisayar Mühendisliği Bölümü, Yaşar Üniversitesi, İzmir, Türkiye

arman.savran@yasar.edu.tr

Öz
Konuşma sesi tespiti (KST), insan bilgisayar arayüzleri için yaygın olarak kullanılan gerekli bir ön-işlemedir. Karmaşık akustik arka plan gürültülerinin varlığı, büyük derin sinir ağlarının ağır hesaplama yükü pahasına kullanımlarını gerekli kılmaktadır. Görü yoluyla KST ise, arka plan gürültüsü problemi olmadığından, tercih edilebilen alternatif bir yaklaşımdır. Görü kanalı, ses verisine erişimin mümkün olmadığı durumlarda ise zaten tek seçenektir. Ancak, genelde uzun süreler aralıksız çalışması beklenen görsel KST, video kamerası donanım ve video verisi işleme gereksinimlerinden dolayı önemli enerji sarfiyatına sebep olur. Bu çalışmada, görü yoluyla KST için, nöromorfik teknoloji sayesinde verimliliği geleneksel video kameradan oldukça yüksek olan olay kamerasının kullanımı incelenmiştir. Olay kamerasının yüksek zaman çözünürlüklerinde algılama yapması sayesinde, uzamsal boyut tamamen indirgenerek sadece zaman boyutundaki örüntülerin öğrenilmesine dayanan son derece hafif fakat başarılı modeller tasarlanmıştır. Tasarımlar, zamansal alıcı alan genişlikleri gözetilerek, farklı evrişim genleştirme tiplerinin, aşağı-örnekleme yöntemlerinin ve evrişim ayırma tekniklerinin bileşimleri ile yapılır. Deneylerde, KST'nin çeşitli yüz eylemleri karşısındaki dayanıklıkları ölçülmüştür. Sonuçlar, aşağı-örneklemenin yüksek başarım ve verimlilik için gerekli olduğunu ve bunun için, maksimum-havuzlamanın adımlı evrişim yöntemiyle aşağı-örnekleme yapmaktan daha üstün başarım elde ettiğini göstermektedir. Bu şekilde üstün başarımlı standart tasarım 1.57 milyon kayan nokta işlemle (MFLOPS) çalışır. Evrişim genleştirmesinin sabit bir faktörle yapılıp aşağı-alt örnekleme ile birleştirilmesiyle de, benzer başarımla, işlem gereksiniminin yarıdan fazla azaldığı bulunmuştur. Ayrıca, derinlemesine ayrışım da uygulanarak işlem gereksinimi 0.30 MFLOPS'a, yani standart modelin beşte birinden daha aşağısına indirilmiştir.

Anahtar kelimeler: Konuşma Sesi Tespiti, Olay Kamerası, Verimli, Görsel Konuşma, Genleştirilmiş Evrişim, Ayrılabilir Evrişim

Temporal Convolutional Networks for Efficient Voice Activity Detection with Event Camera

Abstract
Voice activity detection (VAD) is a widely used essential pre-processing for human-computer interfaces. The presence of complex acoustic background noise requires the use of large deep neural networks at the expense of heavy computational load. Visual VAD is a preferable alternative approach since there is no background noise problem. Also, the video channel is the only option when access to audio data is impossible. However, visual VAD, which is generally expected to operate continuously for long periods of time, causes significant energy consumption due to the requirements of video camera hardware and video data processing. In this study, the use of the event camera, whose efficiency is much higher than the traditional video camera thanks to neuromorphic technology, was examined for VAD through vision. Thanks to the event camera's detection at high time resolutions, the spatial dimension is completely reduced and extremely lightweight but successful models that work only in the time dimension have been designed. Designs are made with combinations of different types of dilated convolution, down-sampling methods, and separable convolution techniques, taking into account temporal receptive field sizes. In the experiments, the robustness of VAD against various facial actions was measured. The results show that down-sampling is necessary for high performance and efficiency, and for this, max-pooling achieves superior performance than down-sampling with stepwise convolution. This high-performance standard design operates at 1.57 million floating point operations (MFLOPS). By performing dilated convolution with a constant factor and combining it with down-subsampling, it was found that the processing requirement was reduced by more than half, with similar performance. Additionally, by also applying depthwise separation, the processing requirement was reduced to 0.30 MFLOPS, less than one-fifth of the standard model.

Keywords: Voice Activity Detection, Event Camera, Efficient, Visual Speech, Dilated Convolution, Separable Convolution

* Sorumlu yazar.
E-posta adresi: arman.savran@yasar.edu.tr

Gönderme : 4 Aralık 2023
Revizyon : 8 Nisan 2024
Kabul : 18 Nisan 2024

1. Giriş (Introduction)

Konuşma sesi tespiti (KST), işitsel veya görsel konuşma tabanlı arayüzler ve sahne analizi için önemli bir ön-işlemedir. Ses-temelli KST, konuşma tanıma, konuşmacı tanıma, konuşma sesi iyileştirme, konuşmacı günlüğü çıkarma, komut-kontrol gibi uygulamalarda (Wang vd., 2023, Korkmaz ve Boyacı, 2023, Zhang vd., 2016, Çubukçu vd., 2015) yaygın kullanılsa da bazı sınırlamaları vardır. Arka plan gürültüsü başlıca bir güçlüktür. Büyük derin sinir ağları (DSA'lar) çeşitli zorlu gürültü koşullarında etkili çözüm sağlayabilse de ağır hesaplama yükü ve enerji tüketimine neden olurlar (Zhang vd., 2016). KST birçok uygulamada ön-işleme olarak çalışması gerektiğinden yüksek verimlilik önem arz eder. Öte yandan, görü yoluyla KST dudak hareketlerinden konuşmayı saptarken akustik gürültünün hiçbir etkisi yoktur. Teknik nedenler veya gizlilik ihtiyacından dolayı işitsel kanalın bulunmaması ise, işitsel VAD için bir kısıtlamadır; bu durumlarda tek seçenek görü yoluyla tespit olabilir. Bahsedilen nedenlerden dolayı, görsel-işitsel veya yalnızca görsel kipte çalışan KST yöntemleri üzerine araştırmalar yapılmaktadır (Ariav vd., 2018, Guy vd., 2020).

Bu araştırmalar video kameralar kullanarak oldukça başarılı sonuçlar elde edebilmişlerdir. Ancak, genelde uzun süreler aralıksız çalışması beklenen görsel KST, video kamerası donanım ve video verisi işleme gereksinimlerinden dolayı önemli enerji sarfiyatına sebep olur. Nöromorfik mühendislik ilkeleriyle geliştirilen ve yeni ortaya çıkan olay kameralarının, geleneksel video kameralara göre üstün olduğu yönleri, KST için daha başarılı ve daha verimli çözümlerin geliştirilmesini sağlayabilir. Olay kamerasının, robotik uygulamalarda, uçan gözlerde (drone), otonom araçlarda veya mobil cihazlarda geleneksel kameralara göre avantajlı bir alternatif olduğu veya video kamerasını tamamlayıcı bir görü kanalı olduğu zaten birçok defa gösterilmiştir (Gallego vd., 2022). Olay kamerasının başarısı, basit bir deyişle, "akıllı piksel" temelli bir algılama tekniğinden kaynaklanmaktadır. "Akıllı piksel" tabiri, üzerine düşen ışık şiddetinin değişimini, diğer piksellere bağlı olmadan kendi başına saptayabilen pikseli ifade eder. Geleneksel kamera piksellerinde böyle bir mekanizma yoktur, sadece ışık yoğunluğu sayısal olarak örneklenir. Bu saptama olayı, bir piksel-olayı olarak kodlanarak arabirim üzerinden aktarılır. Olay kamerasına özgü bu asenkron tetiklenen piksel-olayları sayesinde çok yüksek zamansal çözünürlük, düşük gecikme, düşük güç gereksinimi ve yüksek dinamik aralık gibi önemli avantajlar elde edilir. Bu avantajlar, olay kamerası temelli KST ön-işlemesi sayesinde, daha verimli ve başarımlı yeni nesil görsel veya görsel-işitsel uygulamaların geliştirilmesinin önünü açabilir.

Görsel KST problemi ile yakından ilgili olarak, insan yüzü işleme alanında olay kamerası ile yapılan çalışmalar mevcuttur. Örneğin, olay kamerası kullanılarak gerçekleştirilen otomatik dudak okumanın, geleneksel kamera başarımını geride bırakabileceği gösterilmiştir (Tan vd., 2022). Bu sonuç, dinamik örüntülerin yüksek zaman çözünürlüğü ile algılanması sayesinde gerekli bilginin korunması argümanı ile açıklanabilir. Örneğin, gerçek hayattaki böyle bir uygulamada, bir KST ön-işleme biriminin devreye girmesiyle, gereksiz yere işlemci meşgul eden ve enerji harcayan DSA hesaplamalarının yapılmasının önüne geçilebilir ve konuşma dışındaki görsel aktivitelerin yanlış dudak okumaya neden olması önlenebilir. Başka bir uygulama olarak, örneğin, konuşma dinamikleri öğrenilmesi yoluyla kişi tanıma ele alınabilir. Olay kamerasının konuşma dinamiği temelli kişi tanımadaki avantajları gösterilmiştir (Moreira vd., 2022). Konuşmacı sesi iyileştirme (Arriandiaga vd., 2021) gibi çalışmalarla örnek uygulamalar çoğaltılabilir. KST, benzeri bütün uygulamalarda gerekli bir ön-işlemedir ve ayrıca, akustik ses işleme gibi olay kamerasının kullanılmadığı uygulamalarda da aktivasyon ön-işlemesi olarak görev alabilir.

Bu çalışma, bu tür sistemleri olanaklı kılabilmek için, kaynak gereksinimi çok düşük seviyede olan ve buna rağmen etkili olarak çalışılabilen yöntemlerin geliştirilmesi üzerinedir. Görü verileri, sahnedeki aktiviteye uyumlu olarak yüksek zaman çözünürlüklerinde seyrek yapıda oluştuğundan, sadece zaman boyutundaki değişim örüntülerinin öğrenilmesi yoluyla KST geliştirilmesi hedeflenmiştir. Burada ana fikir, uzamsal boyutu tamamen indirgeyerek işlem yükünden büyük oranda tasarruf edebilmektir. Ağız bölgesindeki piksel-olaylarının uzamsal boyutu tamamen indirgenerek ve sadece zamansal eksende örüntü tanıma yaparak, çok düşük gereksinimli fakat yüksek başarımlı sınıflandırıcılar hedeflenmiştir. Konuşma dinamiği örüntü özniteliklerini öğrenme yoluyla çıkarmak için ise zamansal evrişimli ağlar uygulanmıştır. Bunun sebebi, öz yinelemeli ve dönüştürücü sinir ağlarına kıyasla çok daha verimli çalışmaları ve çok büyük olmayan veri kümelerinde en iyilemesi nispeten kolay olmasıdır (Bai vd. 2018). Farklı evrişim tekniklerini kullanan tasarımlar olay kamerası temelli KST problemi için sınanmıştır. Aşağı-örnekleme, genleştirme, derinlemesine ve gruplamalı ayrılabilir evrişim teknikleri kullanılarak ve zamansal alıcı alanları analiz edilerek tasarımlar yapılmıştır. Zaman ekseninde maksimum havuzlama yoluyla aşağı-örnekleme yapılarak çeşitli yüz eylemlerine karşı gürbüz çalışan ve yüksek başarımlara ulaşan bir standart tasarım önerilmiştir. Evrişim genleştirme ve derinlemesine ayırma yapılarak da bu standart yönteme göre beş kattan fazla işlem kazancı sağlanmıştır.

Makalenin geri kalanında, önce Bölüm 2'de ilgili çalışmalara yer verilir. Bölüm 3'te, uzamsal alanı ağız bölgesinde indirgeyen piksel-olayı temsil modeli ve önerilen çeşitli zamansal evrişim teknikleri ve tasarımları anlatılır. Bölüm 4'te, verimlilik artırma tasarımlarının başarım ve işlem yükleri sunularak başarım ve verimlilik açılarından en iyi modeller saptanır ve konuşma dışındaki yüz dinamiklerine karşı dayanıklıkları ayrı ayrı ölçülür. Son olarak Bölüm 5'te, elde edilen bulgular özetlenerek ana sonuçlar verilir.

2. İlgili Çalışmalar **(Related Work)**

Geleneksel kameralarla yapılan önceki KST çalışmaları akustik ortam gürültüsüne karşı dayanıklılığı artırmak için, birçok defa görsel-işitsel çözümler önermişlerdir (Ariav vd., 2018, Ghaemmaghami vd., 2015). Ancak, işitsel veriler olmadığı durumlarda, KST için tek seçenek video verileri olabilir. Görü yoluyla KST için, Patrona vd. (2016) optik akış ve görüntü gradyanı tanımlayıcıları ile görsel-kelime-çantasına dayalı bir teknik önermiştir. Guy vd. (2020) özyinelemeli ağları doğrudan yüz nirengi noktalarına uygulamış ve optik akış örüntülerini evrişimli ağlar ile modellemişlerdir.

Yakın çekim yüz sahnelerinde yapılan KST çalışmalarının yanında, birden fazla insan vücudunu ve diğer ön plan nesnelerini içeren, arka plan karmaşıklığı fazla olan çok geniş açılı sahnelerde konuşma ile ilgili yüz ve vücut kısımlarını saptayan KST çalışmaları da vardır (Sharma vd., 2019, Shahid vd., 2021). Bu kapsamdaki bütünleşik mekansal lokalizasyon ve KST problemi daha zorlu olduğundan, çok daha karmaşık modellerin kullanılmasını gerektirir. Ek olarak bunların uygulama alanları bu makalenin hedeflerinden daha farklıdır. Dolayısıyla, böyle bütünleşik problemler bu çalışmanın kapsamı dışındadırlar.

Diğer taraftan, nöromorfik sensörler, kendilerini sahne aktivitesine uyarlayarak ve sıkıştırılmış seyrek algılama gerçekleştirerek çok yüksek zaman çözünürlüğü, enerji verimliliği ve yüksek dinamik aralığı avantajları sunarlar (Gallego vd., 2022). Geleneksel sensör, tüm piksellerdeki ışık yoğunluğunu eş zamanlı olarak örneklediğinden bu özelliklerden mahrumdur. Nöromorfik sensör ise, bir piksel üzerindeki ışık yoğunluğunda bir miktar değişiklik tespit ettiği anda, diğer piksellerle eş zamanlı olmayan bir piksel-olayı oluşturur. Bu yeni algılama teknolojisi birçok başarılı uygulamanın geliştirilmesini sağlamıştır. Örneğin, nesne sınıflandırma görevlerinde (Deng vd., 2022, Kim vd., 2022, Schaefer vd. 2022, Gehrig vd., 2019), el hareketi işaretlerinin tanınmasında (Amir vd., 2017), yürüme biçimi tanımada (Wang vd., 2019, Wang vd., 2022), nesne saptamada (Li vd., 2022, Schaefer vd., 2022, Perot vd., 2020) ve izlemede (Zhang vd., 2022), otonom sürüş için direksiyonu dönmesini tahmin etmek amacıyla (Maqueda vd., 2018), kamera poz takibi (Gallego vd., 2018) ve optik akış tahmini için (ParedesValles vd., 2021, Gehrig vd. 2019) kullanılmıştır. Ayrıca, olay kamerası verilerinden video geri-çatım işleminin başarılı bir şekilde yapılabileceği gösterilmiş (Zhu vd., 2022, ParedesValles vd., 2021, Rebecq vd., 2019) ve, standart videolarda hareket bulanıklığını gidermek için olay kamerası kullanımı (Tulyakov vd., 2022, Pan vd., 2019) önerilmiştir.

Konuşma artikülasyonlarının dinamik örüntülerinin taşıdığı bilgi seviyesi son derece yüksek olabildiğinden, olay kamerasının zamansal çözünürlük avantajından faydalanmayı ilke edinen çeşitli konuşma işleme araştırmaları yapılmıştır. Çalışmaların birçoğu, başarımı artırmak için işitsel ve görsel sinyallerin birleşimine odaklanmıştır. Neil vd. (2016) ve Li vd. (2019) görsel-işitsel konuşma tanıma için DSA'lar aracılığıyla görsel olay verilerini ses kipiyle birleştirmiş; Savran vd. (2018) bir görsel uzam-zamansal filtreyi işitsel DSA ile birleştirerek gürültülü akustik ortamlarda konuşma sesi algılayıcı başarım ve verimliliğini artırmış; Arriandiaga vd. (2021) konuşmacıyı ayırarak konuşma iyileştirme yapmak amacıyla yüz nirengi noktalarında optik akış kestirimi yapmıştır. Yakın zamanda, Tan vd. (2022) en güncel video dudak okuma yöntemlerini geride bırakan üstün dudak okuma başarımları elde etmiş ve, Savran (2023a) ses aktivitesi bulma için tamamen evrişimsel DSA önermiştir. Bunların dışında, göz kırpma saptama (Lenz vd., 2020, Ryan vd., 2021), yüz pozu hizalama (Savran ve Bartolozzi, 2020, Savran, 2023, Savran, 2023b), yüz bulma (Barua vd., 2016), kimlik tanıma (Moreira vd., 2022) ve ifade tanıma (Berlincioni vd., 2023) gibi görsel konuşmanın yanı sıra çeşitli olay kamerası temelli yüz işleme çalışmaları da mevcuttur.

3. Yöntem (Methodology)

Önerilen görsel KST, Şekil 1'de gösterilmektedir. Önce piksel-olayları verisi kullanılarak ağız bölgesi olay yoğunluğu kestirimi yapılır, böylece konuşma sesi ile ilgili görsel veriler çok düşük boyutlu ve DSA işlemlerine uygun bir forma indirgenir. Bu temsil biçimi Bölüm 3.1'de anlatılmaktadır. Sonra evrişimsel sinir ağı gövdesinde öznitelikler çıkarılır ve baş kısmında tahmin yapılır. DSA'nın gövde kısmı, Bölüm 3.2 ve Bölüm 3.3'te anlatılan teknikler uygulanarak modellenir. Bölüm 3.2'de, alıcı alanı büyütürken karmaşıklığı fazla artırmayan model mimarileri ve Bölüm 3.3'te de, işlem yükünü azaltan modellerler açıklanmaktadır. Ağ mimari tasarımı ise Bölüm 3.4'te yapılır.

3.1. Görsel Olay Yoğunluğu Çıkarımı (Visual Event Intensity Extraction)

Bir piksel-olayı, sensör pikselindeki logaritmik ışık yoğunluğu değişimi belli bir eşik değerini aştığı anda, diğer sensör piksellerinden bağımsız olarak, yani asenkron olarak, bir tetiklenme sonucu oluşur (Gallego vd., 2022). Piksel-olayı, değişimin pozitif veya negatif yönde olduğunu belirten ikili polarite değişkeni p, sensör düzlemindeki konum (x,y) ve, zaman etikteki t bilgilerini içerir. i indeksli piksel-olayı bir $e^i = (x^i, y^i, t^i, p^i)$ çok-öğelisi ile ifade edilir. Dudaklar ana konuşma artikülatörleri olduğundan, ses aktivitesinin dinamik görsel örüntülerini elde etmek için ağız bölgesindeki piksel olaylarının yoğunluğu zaman ekseni boyunca hesaplanır. Bu temsilin yalnızca iki kanalı vardır; biri pozitif, diğeri negatif değişim kutbu içindir. Böylece, ağız bölgesi üzerindeki uzamsal alan tamamen indirgenerek zaman ekseninde değişen sadece iki boyutlu bir gösterim elde edilir.

Ağız bölgesi, yüz nirengi noktaları kullanılarak çıkarılır. Ağız merkez noktası bu bölgenin merkezi olarak alınır. Dikdörtgen biçiminde bir referans koordinat sistemi üzerinde bölge şablonu tanımlanır. Bu çalışmada, dikdörtgen uzunluk-genişlik oranı ¾ olarak belirlenmiştir. Ancak, ağız bölgesinin iki boyutlu izdüşümü değişen yüz pozu nedeni ile biçim değiştirdiği için, şablonun değişen poza göre hizalanması gerekir. Hizalama için, göz ve ağız merkez noktalarını temel alan 2 boyutlu afin poz kestirimi yapılır (Savran ve Bartolozzi, 2020). Her an için değişen bu afin dönüşüm, dikdörtgen şablonu her seferinde o anki poza uygun bir dörtgene dönüştürür. Böylece, büyük pozlar altında dahi, konuşma eylemi ile en ilgili piksel-olaylarını kullanılması sağlanır. Şekil 2'de örnek ağız bölgeleri gösterilmektedir.

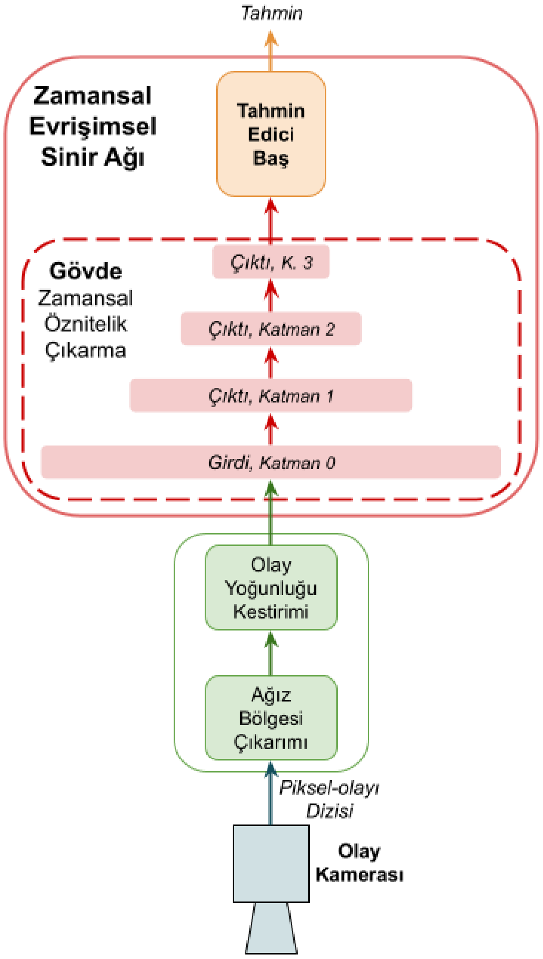
Piksel-olayları yığınlanarak istenen bir zaman çözünürlüğünde olay yoğunluğu örneklenir. Bunun için doğrusal interpolasyona dayanan ve olay dürtü-tensörü olarak da bilinen gösterimin (Gehrig vd., 2019) uzamsal olarak indirgenmiş özelleşmesi, zamanla değişen ağız alanı normalizasyonu yapılarak uygulanır. Ağız alanı normalizasyonu, görüntü düzlemindeki bölgenin poza bağlı büyüyüp küçülme farklılıklarına karşı değişmezlik kazandırmak için yapılır. Böylece, ağız üzerindeki toplam piksel-olayı sayısı yerine, ağız bölgesindeki olay yoğunluğuna göre analiz yapılır. Aşağıda formülü verilen bu gösterimde M ağız bölgesindeki olay indekslerinin kümesini, N toplam olay sayısını, T zamandaki örnekleme için (yani zamansal niceleme için) bir klipte kullanılan zaman selelerinin toplam sayısını, t sele indeksini, A_t de t anındaki ağız bölgesi alanını ifade eder. Bu sayede her klip, ifadesi

$$I \in R^{2\times T} \tag{1}$$

$$I_{p,t} = \frac{1}{A_t}\left(\sum_{i,p^i=p,i\in M} max\left(0, 1-\left|t-t_b^i\right|\right)\right) \tag{2}$$

$$t_b^i = (T-1)\cdot\frac{t^i - t^0}{t^N - t^0} \tag{3}$$

şeklinde olan bir olay yoğunluğu zaman dizisi **I** matrisi ile temsil edilir. Bu çalışmada, 10 ms genişliğinde zaman seleleri üzerinde 10 ms boyunda adımlarla hesaplama yapılarak olay yoğunluğu dizileri hesaplanır.



Şekil 1. Olay kamerası konuşma sesi tespitinin blok şeması (Event camera voice activity detection block diagram)

3.2. Alıcı Alan Genişletme Modelleri (Receptive Field Widening Models)

Evrişimsel sinir ağ tasarımlarında sık kullanılan alıcı alan (receptive field) terimi, sinir biliminden gelmektedir. Alıcı alan, kısaca, sinirsel bir yanıt üretilebilmesi için algı uzayında gereken bölgenin genişliğidir. Evrişimsel ağlarda ise, evrişim katmanlarındaki faydalı öznitelik çıktılarını oluşturabilen girdi sinyalindeki alan genişliği olarak tanımlanır. Alıcı alanın genişliği ve nasıl genişletildiği bir evrişimsel model mimarisinin başarımını belirleyen başlıca faktörlerdendir. Temel mimaride her ne kadar ağın baş kısmında tüm girdi alanı kullanılabilir olsa da, gövdede oluşturulan zamansal öznitelikler kısıtlı bir alandaki örüntüleri yakalayabilmektedirler. Dolayısıyla, zamansal alıcı alanı etkili ve aynı zamanda verimli bir şekilde genişleterek kullanışlı öznitelikler çıkarabilmek için farklı yöntemlerin incelenmesi gerekir.

En basit olarak, evrişim çekirdeğinin genişliğini artırmak doğrudan alıcı alanı genişletir. Ancak çekirdek genişliğindeki artış alıcı alan genişliğindeki artışa eşit olacağından, bu yöntem son derece verimsizdir ve aşırı sayıda parametre (sinir bağlantı ağırlıkları) gerektirmesi öğrenme problemini çok zorlaştırır. Bundan dolayı bu yöntem hiçbir çalışmada kullanılmamıştır. Aksine modern mimariler en küçük çekirdek genişliği ile derinliği artırmayı ilke edinmişlerdir. Daha derin ağlar, daha çok aktivasyon fonksiyonunu da kullanmayı sağladığından, öğrenmenin başarısını artıran doğrusalsızlık derecesini de yükseltirler. L sayıda evrişim katmanına sahip bir mimaride, eğer her katmandaki çekirdek genişliği k ise, en son katmanın girdi katmanı üzerindeki alıcı alanının genişliği

$$RF = 1 + (k-1)L \tag{4}$$

formülü ile hesaplanır. Denklem 4'ten görüldüğü üzere katman sayısı ve alıcı alan arasında faktörü *k-1* olan doğrusal bir ilişki vardır. Daha hızlı, yani daha az sayıda katman ve parametre ile, benzer alıcı alan genişliklerine aşağı-örnekleme yoluyla ulaşılabilir. Evrişimsel ağlarda, maksimum-havuzlama ve adımlı-evrişim olmak üzere iki yaygın aşağı-örnekleme yöntemi kullanılır. Adımlı-evrişimde, evrişim çekirdeği kaydırılırken standart bir birimlik kaydırma yerine daha büyük bir tamsayı kaydırma miktarı kullanılır. Maksimum-havuzlamada ise, çok ufak bir alanda en büyük değeri seçen bir doğrusal-olmayan ara hesaplama katmanı eklenir ve havuzlama penceresi de birden büyük tamsayı bir adımla kaydırılır. Katmandan katmana değişen çekirdek ve adım genişlikleri ile genelleştirilmiş alıcı alanın kapalı çözümü

$$RF = 1 + \sum_{l=1}^{L}\left((k_l - 1)\prod_{i=1}^{l-1} s_i\right) \tag{5}$$

olduğu gösterilebilir (Araujo vd., 2019). Burada s, her iki yöntemdeki adım boyudur. Sabit k ve s için, birden büyük adımlar olduğunu varsayarsak, aşağıdaki formülü toplam serisi kuralına göre elde ederiz.

$$RF = 1 + (k-1)\frac{s^L - 1}{s - 1} \tag{6}$$

Alıcı alanı hızla büyütmek için daha farklı bir yol ise genleştirilmiş evrişim uygulamaktır (Yu ve Koltun, 2016). Genleştirilmiş evrişimin standarttan farkı, bir çıktı noktasındaki evrişim yanıtını hesaplarken belli girdi noktalarının atlanarak çekirdek çarpımının yapılmasıdır. Genleşme faktörü d olsun. O zaman ardışık her girdi noktası çifti arasında *d-1* nokta atlanarak evrişim uygulanır. Genleştirilmiş evrişimde çekirdek genişlemez fakat girdi sinyali üzerinde kapsadığı sınırlar, bir nevi her araya *d-1* tane delik konularak suni olarak genişletilmiş olur. Dolayısıyla bir çıktı noktasının hemen altındaki girdi katmanında kapladığı alan, standart evrişimdeki k yerine,

$$k' = 1 + (k-1)d \tag{7}$$

olur. Dolayısıyla, eğer Denklem 4'te yerine koyarsak alıcı alanı

$$RF = 1 + (k-1)dL \tag{8}$$

şeklinde elde ederiz. Literatürde genleştirmeyi daha da hızlı yapabilmek için tercih edilen bir yöntem de genleştirmeyi üstel olarak yapmaktır (Rethage vd., 2018). Genleştirme faktör tabanına m dersek, katman endeksine göre genleştirme faktörü $d = m^{l-1}$ olur. Bu ifadeyi Denklem 7'de yerine koyup sonucunu da Denklem 5'te yerine koyarsak, adım genişliğini $s = 1$ olduğu takdirde, toplam serisi uygulamasıyla alıcı alan formülü

$$RF = 1 + (k-1)\frac{m^L - 1}{m - 1} \tag{9}$$

olarak elde edilir. Ayrıca, birden farklı adım genişliği için alıcı alanın

$$RF = 1 + (k-1)\frac{(sm)^L - 1}{sm - 1} \tag{10}$$

olduğu gösterilebilir. Bölüm 3.4.'te uygun tasarım parametreleri saptanarak, burada gösterilen çeşitli alıcı alan genişletme yöntemleri Bölüm 4.3.'te verimlilik ve başarım açılarından değerlendirilirmiştir.

3.3. Derinlemesine ve Gruplamalı Ayrılabilir Evrişim Modelleri (Depthwise and Groupwise Separable Convolution Models)

Derinlemesine ayrılabilir evrişim yöntemi, MobileNets (Howard vd., 2017) ile yaygın kullanımı ortaya çıkan, çok çeşitli görevlerde yüksek verimlilik kazandırdığı gösterilen bir yöntemdir. Burada kullanılan derinlemesine terimi kanal ekseni üzerinde yapılan anlamına gelir. Bu yöntem ile, çok kanallı evrişimler tek kanallı evrişimlere ayrılma yoluyla faktörize edilerek, kanal sayısıyla orantılı işlem tasarrufu sağlanır. İki aşamada gerçekleşir. İlk olarak, her bir girdi kanalına özgü ayrı bir filtre uygulanır. Bu ilk işlem derinlemesine evrişimdir ve bu aşamada girdi kanalları arasındaki herhangi bir etkileşim örüntüsü öğrenilmez. İkici aşamada ise, tam tersi şekilde, kanallar arasındaki etkileşimi öğrenen fakat girdi alanı üzerindeki örüntüleri algılamayan noktalamasına evrişim (pointwise convolution) uygulanır. Noktalamasına evrişim, girdi alanında sadece bir birim kaplayan evrişimdir. Bu evrişim filtresi, sadece tek bir girdi noktasındaki kanal değerlerinden hedeflenen çıktı kanal sayısını oluşturacak şekilde ayarlanır. Böylece, normalde çok kanallı girdi ve çıktılar için uygulanan tek bir büyük filtre yerine, çok daha az parametre sayısına sahip olan kanala özgü filtreler ve ardından da, kanalların doğrusal birleşimini gerçekleştiren noktalamasına filtre uygulanmasıyla aynı görevin daha verimli bir şekilde yerine getirilmesi sağlanır.

Bir evrişim katmanında filtre genişliği *K*, girdi kanal sayısı *M* ve çıktı kanal sayısı *N* olsun. O zaman standart yöntemde toplam evrişim parametre sayısı $M \times K \times N$ olur. Eğer çıktı alan genişliği de T ise, evrişim hesaplama yükü de $M \times K \times N \times T$ olur. Derinlemesine evrişimde ise, $M = N$ olur ve her kanal için ayrı filtre uygulandığından $M \times K$ parametre gelir. Noktalamasına filtreden dolayı da $M \times N$ parametre olduğundan, sonuçta toplam parametre sayısı $M \times (K + N)$ olur. Dolayısıyla derinlemesine evrişimin toplam hesaplama yükü $M \times (K + N) \times T$ olarak bulunur. Böylece hesaplama azalma oranı

$$\frac{M\times(K+N)\times T}{M\times K\times N\times T} = \frac{1}{N} + \frac{1}{K} \tag{11}$$

şeklindedir. Filtre genişliği K genelde ufak sabit bir değerdir. Tipik olarak *K*=3 değeri için, iki üç kat arasında bir hesaplama kazancı elde edilir.

Kanallar üzerinde faktörizasyon yaparken diğer bir seçenek de her bir kanal yerine, kanalları gruplandırıp her bir kanal grubu için bir filtre kullanmaktır, yani gruplamalı evrişim uygulamaktır (Krizhevsky vd., 2012). Derinlemesine evrişimle aynı şekilde yine ikinci aşamada noktalamasına evrişim uygulanarak ayrılabilir gruplamalı evrişim gerçekleştirilir. Dolayısıyla, *G* tane grup kullanıldığında gruplamalı evrişim toplam parametre sayısı

$$G \times M/G \times M/G \times K = M^2 \times K/G \tag{12}$$

ve noktalamalı evrişim toplam parametre sayısı $M \times N$ olduğundan, toplam parametre sayısı $M \times \left(\frac{M\times K}{G+N}\right)$ olur. Böylece, Denklem 11'de yaptığımız gibi hesaplama azalma oranı

$$\frac{M\times(M\times K/G+N)\times T}{M\times K\times N\times T} \tag{13}$$

$$= \frac{M}{G\times N} + \frac{1}{K} \tag{14}$$

olarak bulunur. Burada grup sayısını artırarak hesaplama kazancını artırabildiğimiz ve kanal sayısını *M*'ye eşitlersek de en fazla kazanç olan Denklem 1'deki sonucu elde ettiğimiz görülür. Hesaplama yükünü azaltırken başarım düşebileceğinden, Bölüm 4.4'te farklı grup sayılarına bakılarak hesaplama ve başarım değerleri incelenmiştir.

3.4. Ağ Mimarileri (Network Architectures)

Şekil 1'de gösterilen gövde ve tahmin edici baş kısımları için farklı tasarımlar ele alınabilir. Tahmin edici baş kısmında, gövdede elde edilen zamansal öznitelikler üzerinden sınıflandırma gerçekleştirilir. Bu tür görevler için sıklıkla çok-katmanlı algılayıcı kullanılır. Ancak, çok-katmanlı algılayıcılara gerek olmadan yüksek başarımların elde edilebileceği gösterilmiş ve yaygınlaşmıştır (Szegedy vd., 2015). Bu tür modellerde önce, bütünsel ortalama havuzlama yoluyla çok kanallı olan bütün alan kanal sayısını değiştirmeden indirgenir ve sonra, doğrusal katman uygulanarak istenen hedefler tahmin edilir. Dolayısıyla çok katmanlı modellere göre oldukça basittir ve hiper-parametre gerektirmez. Bu çalışmada, bu en basit tahmin edici baş tasarımının yüksek KST başarımı elde ettiği görüldüğü için karmaşıklığı daha yüksek olan çok-katmanlı tasarımların kullanılmasına gerek duyulmamıştır.

Gövde tasarımında ilke olarak, her bir yukarı katmanda bir birimdeki öznitelik miktarını yani kanal sayısını artırmak ve böylece daha zengin bir temsil elde etmek hedeflenirken, aşağı-örnekleme ile de aktivasyon hacminin aşırı büyümesinin önlenmesi hedeflenmiştir; çünkü aktivasyon hacminin fazla büyük olması işlem yükünü ciddi oranda artırmaktadır. Bu uygulama, sayısal bilgisayar donanımı için elverişli olan ikili sisteme göre

$$c_l = 2^{1+l} \tag{15}$$

parametrizasyonu ile yapılır. Burada, *l* katman indisidir ve girdi katmanı da zaten iki kanallı olduğundan $c_0 = 2$'dir. Evrişim, genişliği üç olan çekirdekler kullanıldığında aktivasyon hacmini ikinin katları şeklinde olmasını sağlamak amacıyla, bir birim genişliğindeki sıfır-dolgulama ile yapılır. Her katmanda, evrişimden sonra ReLU aktivasyonu uygulanır.

Aktivasyon alanını indirgemek için, Bölüm 3.2.'de anlatılan farklı alıcı alan genişletme modelleri uygulanır. Bu çalışmada, SE olarak adlandırdığımız standart model, aşağı-örnekleme indirgeme faktörü, yani adım genişliği *s*=2, ile maksimum-havuzlama yapılan model anlamına gelmektedir. Maksimum-havuzlama yerine doğrudan adımlı evrişim ile aynı adım genişliği, literatürde tamamen evrişimsel, yani TE modeller ile yaygındır (Long vd., 2015). Bölüm 4.4'te bu modeller karşılaştırılır. Bu iki modelin algısal alan genişlikleri Denklem 6 ile hesaplandığında, *k*=3 olduğundan, katman sayısına bağlı olan formül

$$RF = 1 + (3-1)\frac{2^{L}-1}{2-1} = 2^{L+1} - 1 \qquad (16)$$

olarak bulunur. Aşağı-örnekleme yapmayıp, sadece üstel genleştirme yaptığımızda (ÜGE), eğer Denklem 9'da m=2 olarak alırsak, yine Denklem 16'daki aynı alıcı alan ilişkisini elde ederiz.

Genleştirme yolu ile alıcı alanı genişletip ve aynı zamanda aşağı-örnekleme yapan iki farklı genleştirme modeli uygulanır. Bunlar sabit genleştirme ve üstel genleştirme modelleridir. Genleşmiş filtre genişliği Denklem 7 ile hesaplanır. Böylece, sabit genleştirme ve aşağı-örnekleme için (GSE), d=2, s=2 ve, k=3 olduğundan, Denklem 6 ile ifade edilen alıcı alan formülüne göre

$$k' = 1 + (3-1)2 = 5 \qquad (17)$$

$$RF = 1 + (5-1)\frac{2^{L}-1}{2-1} \qquad (18)$$

$$= 2^{L+2} - 3 \qquad (19)$$

bulunur. Alıcı alanı artan katman sayısıyla çok daha hızlı büyüten üstel genleştirme ve aşağı-örnekleme için (ÜGSE) ise, m=2 için $d = 2^{l-1}$ olur ve Denklem 10'da m=2, s=2 ve k=3 için

$$RF = 1 + (k-1)\frac{(2\cdot 2)^{L}-1}{2\cdot 2-1} \qquad (20)$$

$$= (1 + 2^{2L+1})/3 \quad . \qquad (21)$$

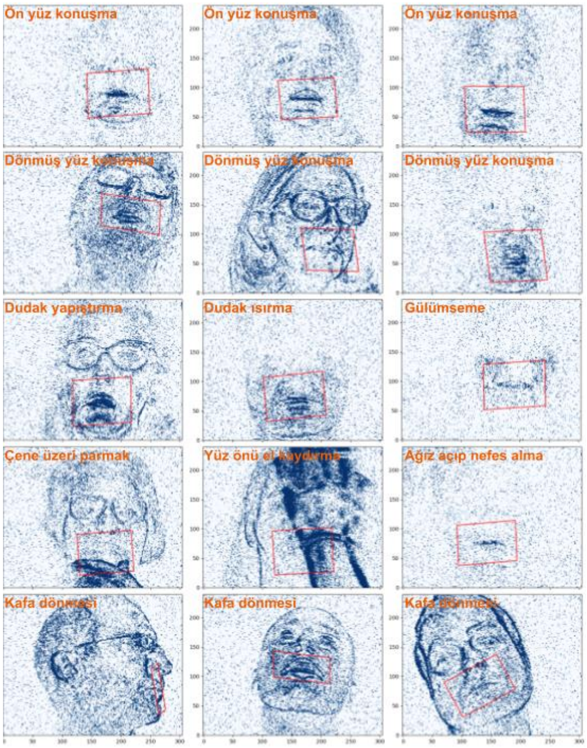
Tablo 1. Evrişimsel (E.) modellerin kısaltmaları (Abbreviations of the convolutional (E.) models)

Kısaltma	Açılım
SE	Standart Evrişimsel
TE	Tamamen Evrişimsel
ÜGE	Üstel Genleştirilmiş E.
ÜGSE	Üstel Genleştirilmiş Standart E.
GSE	Sabit Genleştirilmiş Standart E.
ASE	Ayrılabilir Standart E.
GASE	Sabit Gen. Ayrılabilir Standart E.

Diğer taraftan, verimliliği daha da artırabilmek için, Bölüm 3.3'te anlatılan derinlemesine ayrılabilir ve gruplamasına ayrılabilir evrişimsel tasarımları uygulanır. Gruplamasına evrişim uygularken, ikinin katları şeklinde üstel olarak farklı grup sayıları ile tasarımlar değerlendirilir. Ayrıca, en verimli modeli araştırırken, genleştirme ve ayırma teknikleri birleştirilerek, genleştirilmiş ayrılabilir evrişim en fazla verimlilik için incelenmiştir. Tablo 1'de, bu çalışmada değerlendirilen bütün tasarımlar listelenmektedir. Bölüm 4'te, burada anlatılan bütün tasarımlar ile karşılaştırmalı değerlendirmeler sunulur.

4. Deneyler (Experiments)

Bu bölümde önce deneylerde kullanılan veri kümesi Bölüm 4.1'de anlatılır ve eğitim parametreleri Bölüm 4.2'de verilir. Bölüm 4.3'te farklı alıcı alan modellerinin başarımları ve Bölüm 4.4'te farklı verimlilik artırma modellerinin başarımları karşılaştırılarak incelenir. Bölüm 4.4'te ise modellerin başarım karakteristikleri detaylı bir şekilde analiz edilir.



Şekil 2. Örnek anlık histogram görselleştirmeleri ve hesaplanan ağız bölgeleri (Example histogram visualization snapshots and computed mouth regions)

4.1. Veri kümesi (Dataset)

Deneylerde kullanılan veri kümesi MHz mertebelerindeki zaman çözünürlüğüne erişen ve 304 x 240 piksel çözünürlüğüne sahip bir olay kamerası ile oluşturulmuştur (Savran vd., 2018). Toplam 486 tane klip vardır. Kliplerin 324 tanesi konuşma sesi içermekte, geri kalan 162 tanesi ise içermemektedir. Sesli klipler konuşma işleme çalışmalarında sıklıkla kullanılan fonetik olarak zengin TIMIT (Wrench 2006) metinlerinden oluşmuştur. Konuşma kliplerinin 54 tanesinde kafa dönmesi hareketleri de vardır ve kişiler açılı pozda konuşmuşlardır; diğer konuşma klipleri ön yüz şeklindedir. Konuşmasız klipler, konuşmaya benzeyen ağız hareketlerini de içeren çeşitli eylemler içermektedirler. Bunlar dudak yapıştırma şeklinde açıp kapama, dudak ısırma, gülümseme, çene üstüne parmakla dokunma, eli yüz üzerinde gezdirme ve ağız açarak nefes alma gibi yüz kapatma hareketleri ve, üç eksende farklı hız ve tekrarlama sayılarıyla yapılan kafa dönmesi hareketleridir. Bu eylemleri içeren örnekler Şekil 2'de gösterilmektedir. 18 kişiden toplanan veri kümesinde, klip sayı ve tipleri aynıdır ancak içerikler farklıdır. Konuşmalı kliplerde TIMIT cümleleri kişiden kişiye değişmektedir ve konuşmasız kliplerde kişiler belirtilen eylemleri istedikleri gibi gerçekleştirmişlerdir. Bütün kliplerde ses kanalı görsel kanal ile senkronize edilmiştir. Ayrıca, kliplerin göz ve ağız merkezleri işaretlenmiştir. Çeşitli pozitif ve negatif örnek klipler ses dalga biçimleri ve olay yoğunluk grafikleri ile Şekil 3'te gösterilmektedir. Deneylerde iki kişi geçerleme ve dört kişi test kümesi için ayrılmıştır.

4.2. Eğitim Parametreleri (Training Parameters)

Kullanılan öğrenme kayıp fonksiyonu ağırlıklı ikili-çapraz-entropi-logits fonksiyonudur. Burada ağırlıklandırma, pozitif ve negatif sınıf örnek miktarlarının öğrenmedeki etkilerini dengeleyebilmek için, kayıp fonksiyonunu hesaplarken sınıf örnek sayısına ters orantılı ağırlık çarpanı uygulanarak yapılır. Ayrıca, olay yoğunluğu girdi kanalları üzerinde standart normalizasyon yapılır. Eğitimler için, 32 kliplik yığınlar üzerinden ADAM en iyilemesi 10^{-3} sabit öğrenme oranı ile çalıştırılır. Farklı uzunlukta klipler ile bir yığın tensörü oluşturabilmek için sıfır-dolgulama yapılarak sabit 1024 örnekli klipler elde edilir. Her bir örnek 10 milisaniyeye denk düştüğünden, sabit klip uzunluğu yaklaşık 10 saniyedir. Eğitimler, ayrılabilir evrişim modelleri haricinde varsayılan olarak sabit 50 devirlik döngülerle yapılmıştır; ayrılabilir evrişim modellerinde ise, yakınsamanın daha uzun sürdüğü görüldüğünden 100 devirlik döngülerle yapılmıştır.

4.3. Alıcı Alan Genişletme Modellerinin Değerlendirilmesi (Evaluation of the Receptive Field Widening Models)

Tablo 2'de farklı evrişimsel modellerin katman sayılarına göre değişen alıcı alanları gösterilmektedir. SE ve ÜGE'nin alıcı alanlarının eşit olduğu görülmektedir. Yani sadece aşağı-örnekleme yapıldığında (SE) ve aşağı-örnekleme olmadan üstel genleştirme yapıldığında (ÜGE) alıcı alanlar eşit çıkmaktadır. Bunun nedeni, Bölüm 3.4'te anlatılan tasarımların Denklem 19'daki aynı ilişkiye varmasıdır. Diğer taraftan, sabit genleştirme ile beraber aşağı-örnekleme yapıldığında (GSE), Denklem 22'den dolayı yaklaşık iki kat daha fazla alan genişliği elde edilir. Üstel genleştirme ile beraber yapıldığında (ÜGSE) ise, Denklem 24'ten dolayı çok daha hızlı bir büyüme gerçekleşir.

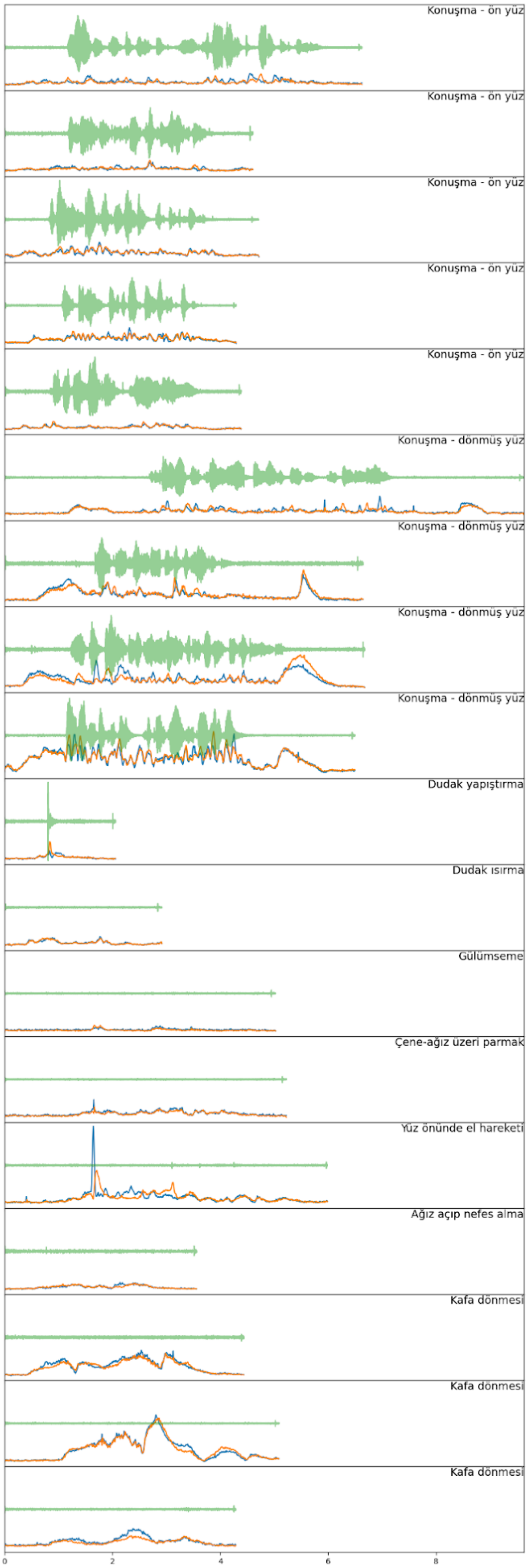
Tablo 2. Alıcı alan genişletme modellerinin alıcı alanları (Receptive fields of the receptive field widening models)

Kat	SE	ÜGE	ÜGSE	GSE
1	3	3	3	5
2	7	7	11	13
3	15	15	43	29
4	31	31	171	61
5	63	63	683	125
6	127	127	2731	253
7	255	255	10923	509

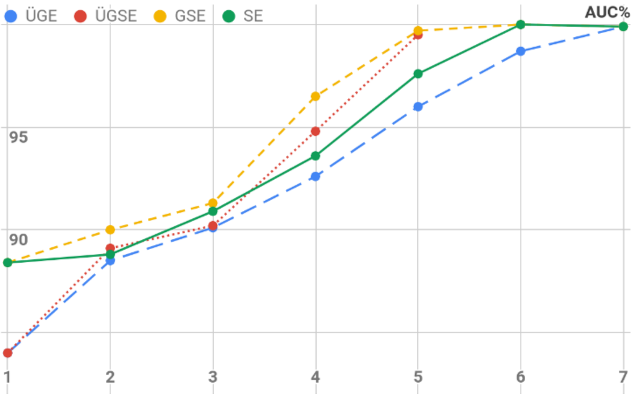
Tablo 3. Alıcı alan genişletme modellerinin MFLOPS çarpma-toplama yükleri (MFLOPS multiplication-addition loads of the receptive field widening models)

Kat	SE	ÜGE	ÜGSE	GSE
1	0.03	0.03	0.03	0.03
2	0.08	0.13	0.08	0.08
3	0.18	0.54	0.18	0.18
4	0.39	2.11	0.35	0.38
5	0.78	8.15	0.51	0.75
6	1.57	30.63	-----	1.44
7	3.15	107.9	-----	2.62

Tablo 3'te de, Tablo 2'deki alıcı alanları listelenen modellerin toplam çarpma-toplama kayan nokta hesaplama yükleri, milyon birimiyle, MFLOPS adı altında, gösterilir. ÜGE modelinde aşağı-örnekleme olmadığı için, Denklem 18'teki ilişkiye göre artan kanal sayısıyla beraber aktivasyon hacmi çok büyür; dolayısıyla da, işlem yükünün katman sayısıyla beraber çok hızlı bir şekilde arttığı görülmektedir. Diğer üç model ise benzer seviyelerde işlem yüküne sahiptir. Artan kanal sayısıyla SE, ÜGSE ve, GSE modellerinin işlem yüklerinde farklılıkların gözlemlenmesinin sebebi, genleştirilmiş modellerde alıcı alanların deneylerde kullanılan 1024 tensör uzunluklarına göre fazla büyümesi ve kırpılma olmasıdır. Bu büyüme ve kırpılmanın, ÜGSE ile altıncı ve yedinci katmanlarda aşırı olmasından dolayı, ÜGSE'nin o katman sayıları deneylere dahil edilememiştir; zaten o kadar büyük alıcı alan kullanmaya zorlamak fazla gecikmeye neden olacağından pratikte kullanışlı da değildir.



Şekil 3. Pozitif ve negatif örnek kliplerin ses dalga biçimleri ve olay yoğunlukları ile gösterimi (Example positive and negative clips shown by audio waveforms and event intensities)



Şekil 4. Alıcı alan genişletme modellerinin değişen katman sayılarına göre (yatay eksen) AUC başarım yüzdeleri (dikey eksen) (AUC percentage performance (vertical axis) of the receptive field widening models with varying layer counts (horizontal axis))

Modellerin başarımları, Şekil 4'te alıcı çalışma karakteristiği (ROC) eğrisi altında kalan alan (AUC) ölçümü ile karşılaştırılır. ROC eğrisi, her bir yanlış pozitif oranına karşı doğru pozitif oranını olası bütün karar eşiği değerleri için verdiğinden, AUC, eşik değeri seçiminden bağımsız olarak elde edilebilecek teorik doğru sınıflandırma başarımını belirtir. Şekil 4'te, GSE ve ÜGSE modellerinin, altı katmana kadar SE'den daha iyi başarım elde ettiği, ÜGE'nin de her katman sayısı için en kötü başarımı elde ettiği görülür. Bu sonuç, alıcı alanın genişliğinin büyük olmasının avantajlı olduğunu göstermektedir (Tablo 2'de alıcı alanlara bakınız). ÜGE'nin işlem yükü gereksinimi de diğer modellere göre çok daha yüksektir. Bu derece keskin başarım ve işlem yükü dezavantajları, aşağı-örnekleminin bu KST probleminde çok önemli olduğuna dair kuvvetli kanıt teşkil eder. Her katmanda GSE'nin başarımı ÜGSE'nin başarımından biraz daha yüksektir. Bu da, sinir ağı derinliği ile artan alıcı alanın gereğinden çok fazla artmasının öğrenmeyi olumsuz etkilediğine işaret eder.

En yüksek başarımlar altı katman ile elde edilmiştir ve, yedi katman kullanıldığında az bir miktar başarım düşüşü gözlemlenmiştir. Altı katman derinlikte, %100 AUC başarımına sahip SE ve GSE modelleri, sırasıyla 127 birimlik ve 253 birimlik alıcı alana sahip olurlar. Düşük alıcı alanla çalışmak daha düşük gecikme anlamına geldiğinden bir avantajdır. Diğer taraftan, beş katman derinliğinde, %99.7 AUC ile GSE açık olarak SE'den daha iyi başarım elde eder ve 125 birimlik bir alıcı alana ihtiyaç duyar. Daha düşük katman sayısı ile yüksek başarım elde ettiğinden GSE'nin işlem yükü de altı katmanlı SE'ye göre daha azdır. Altı katmanlı SE 1.57 MFLOPS ile çalışırken, beş katmanlı GSE 0.75 MFLOPS ile neredeyse aynı alıcı alan ihtiyacı ile çalışır. Bu nedenlerden ötürü, en yüksek başarım istenildiği görevlerde altı katmanlı SE modelinin, fakat verimliliğin kritik olduğu ve az bir miktar başarım düşüşünün tolere edilebileceği koşullarda beş katmanlı GSE modelinin kullanılması uygun olacaktır.

4.4. Ayrılabilir Evrişim Yoluyla Verimlilik Artırma (Efficiency Improvement via Separable Convolution)

Bu bölümde, işlem yükünü hafifleterek verimliliği daha da yukarı seviyelere çıkarmak amacıyla, ayrılabilir evrişim tekniğinin kullanılması değerlendirilir. Hedefimiz düşük işlem yükü olduğu için, Bölüm 4.3'te yüksek başarım seviyesinde en iyi verimliliği sağladığı saptanan beş katmanlı GSE modelinin verimliliğinin daha da artırılması için deneysel inceleme yapılır. Tablo 4'te, derinlemesine ve gruplamasına ayrılabilir evrişim modellerinin MFLOPS işlem yükleri ve AUC başarım yüzdeleri sunulmaktadır. Genleştirilmiş ayrılabilir modeli ifade eden GASE-# biçiminde # sembolü ya gruplamalı yapıdaki grup sayısını ya da Der kısaltmasıyla derinlemesine ayrılabilir evrişimi gösterir. Grup sayısı ikinin katları şeklinde artırılmıştır. Tablo 4'te, grup sayısı artarken işlem yükünün azaldığı fakat azalma hızının grup sayısıyla beraber yavaşladığı görülür. Bunun sebebi alt katmanlardaki düşük kanal sayısı ama büyük aktivasyon alanıdır. Örneğin ilk katmanda iki kanal vardır, dolayısıyla en fazla ikiye bölünebilir. Yukarı katmanlardaki bölünmelerin işlem kazancı ise azalan aktivasyon alanından dolayı nispeten azdır. Diğer bir ifadeyle, alt katmanlarda kanalları ayırmak üst katmanlardakine kıyasla bize daha çok işlem tasarrufu sağlar; çünkü alt katmanlardaki aktivasyon alanları aşağı-örneklemeden dolayı çok daha büyüktür.

Tablo 4. Modellerin (GASE-#, #: grup sayısı veya derinlemesine ayrışım) işlem yükleri ve başarımları (Computation loads and performances of the models (GASE-#, #: group count or depthwise separation)

Model	MFLOPS	AUC %
GSE	0.75	99.7
GASE-2	0.46	99.7
GASE-4	0.37	99.3
GASE-8	0.33	99.1
GASE-16	0.31	98.6
GASE-32	0.30	99.8
GASE-Der	0.30	99.7

GSE'nin yüksek AUC başarım yüzdesi genelde oldukça korunduğu görülmektedir. En düşük işlem yükü derinlemesine ayrılabilir model (GASE-Der) ile elde edildiğinden ve GSE ile aynı başarımı yakaladığı gözlemlendiğinden, en verimli model olarak seçilir. 0.75 MFLOPS işlem yükü, iki kattan fazla düşerek 0.30 MFLOPS seviyesine inmiştir. Ayrılabilir evrişim tekniği SE ve diğer modellerde de verimlilik kazandırır. Ancak ayrılabilir SE'nin ulaştığı başarım, GSE'nin başarım değerlerine ve hatta daha düşük seviyelere düştüğü gözlenmiştir ve buna rağmen işlem yükü daha fazladır.

4.5. Başarım ve Verimlilik Karşılaştırmaları (Comparisons of Performance and Efficiency)

Bu bölümde, en yüksek başarım için seçilen altı katmanlı SE modeli ve en yüksek verimlilik için seçilen beş katmanlı GASE modeli, KST problemi için detaylı olarak karşılaştırılır. Ayrıca, SE modeliyle aynı nitelikte olan fakat tek farkı aşağı-örneklemeyi maksimum-havuzlama yerine doğrudan adımlı evrişim ile yapan TE modeli de incelemeye dahil edilir. Tablo 5'te bu üç modelin MFLOPS işlem yükleri ve AUC başarım yüzdeleri gösterilmektedir. TE modeli 0.79 MFLOPS ile SE'den iki kat daha verimli çalışmakta fakat %98.2 AUC ile en düşük başarımı elde etmektedir. İki kat verimlilik, evrişimin iki birimlik adımla gerçekleşmesi sayesindedir; oysa maksimum havuzlamadan önce yapılan evrişim tek adımlı olur. Tamamen evrişimli sinir ağının bu problemde geride kalmasının sebebi, maksimum-havuzlama ile daha kuvvetli öteleme değişmezliği elde edilmesi olabilir; çünkü, konuşma aralıkları son derece değişken olarak ortaya çıkar. Tablo 5'te en verimli GASE modelinin, sadece %0.3'lük bir başarım kaybına uğrayarak 0.30 MFLOPS ile işlem yükünü SE'nin beşte birinden daha azına indirdiği görülmektedir.

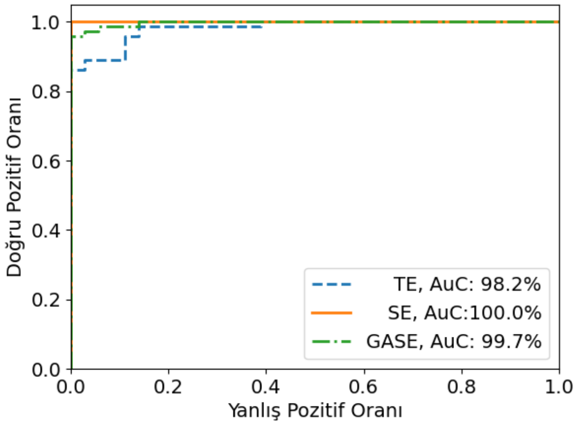
Tablo 5. Tamamen (TE), standart (SE) ve, sabit genleştirilmiş derinlemesine ayrılabilir evrişimsel (GASE) modellerinin işlem yükleri ve başarımları (Computation loads and performances of the fully (TE), standard (SE) and, dilated depthwise separable convolutional (GASE) models)

Model	MFLOPS	AUC %
TE	0.79	98.2
SE	1.57	100.0
GASE	0.30	99.7

Şekil 5'te bu üç modelin ROC eğrileri karılaştırılmaktadır. SE'nin ideal ROC eğrisine sahip olduğu, TE'nin sesi (pozitif) %100 doğru saptama başarısını ancak %50'ye yakın yanlış pozitif oranlarında elde ettiği görülür. GASE'nin doğru pozitif oranları ise SE'ye yakındır ve %100'e çabuk ulaşır.

Bu üç modelin farklı kategorideki kliplerdeki başarımları, karar eşik değerine bağımlı iki farklı çalışma koşulu altında detaylı olarak Tablo 6'da gösterilmektedir. Bu koşullar, tüm test kümesinde en az %80 doğru negatif sağlama ve en az %99 doğru pozitif sağlama koşullarıdır. İlk koşul bize, konuşma sesi olan klipleri saptama başarımını en fazla %20 yanlış alarm toleransı ile değerlendirme imkanı verirken, ikinci koşul ise neredeyse kusursuz sesli klip saptama yaparken yanlış alarm vermeme başarımını, yani ses aktivitesi dışındaki yüz aktivitesi ve eylemlerine karşı gürbüz olma başarımını, daha iyi analiz etmemizi olanaklı kılar. Pozitif örnekler, ön yüzlü ve dönmüş yüz pozlu konuşma kategorilerine ayrılmıştır. Negatif örnekler ise, konuşma içermeyen fakat kafa dönmeleri, dudak açma-kapama ısırma, ağız açıp nefes alma, gülümseme, çene veya ağız üzerinde parmak tutma ve, yüz önünde eli hareket ettirme kategorileridir.

Tablo 6'da, SE'nin her kategoride kusursuz çalıştığı görülmektedir. GASE, en az %80 doğru negatif koşulu altında, %100 doğru pozitif ve %86.1 doğru negatif elde etmiştir. TE de aynı doğru negatif oranında kalmış fakat %98.6 doğru pozitif elde edebilmiştir. Ancak TE'nin ön yüzlü konuşmalarda kusursuz tanıma başarımı gösterdiği fakat dönmüş yüzlerde hata yaptığı görülür. En az %99 doğru pozitif oranı altında ise GASE yine %86.1 doğru negatif yüzdesini elde ederken, TE'de bu oran çok büyük bir düşüşle %61.1'e inmiştir. TE'nin açık olarak konuşma dışındaki aktivitelere genelde başarılı olmadığı görülmektedir, yani tasarımda maksimum havuzlama kullanmadan yüksek KST başarımı elde etmek mümkün olmamaktadır. Negatif örneklere bakıldığında, genel olarak verimliliği yüksek olan GASE modelinin en çok hata yaptığı kategorilerin dudak yapıştırma-ısırma, çene-ağız üzeri parmak ve, yüz önünde el kaydırarak kapatma kategorileri olduğu görülmektedir. Kafa dönmelerinde %90 üzeri başarı ile daha iyi ayırt etmekte, gülümseme ve ağız açarak nefes alma kategorilerinde ise hata yapmadığı gözlemlenmektedir.



Şekil 5. Tamamen (TE), standart (SE) ve, genleştirilmiş derinlemesine ayrılabilir evrişimsel (GASE) modellerinin ROC eğrileri (ROC curves of the fully (TE), standard (SE) and, dilated depthwise separable convolutional (GASE) models)

Tablo 6. En az %80 doğru negatif koşulu ve en az %99 doğru pozitif koşulları altında, tamamen (TE), standart (SE) ve, genleştirilmiş derinlemesine ayrılabilir evrişimsel (GASE) modellerinin kategoriler için doğru pozitif (P) ve doğru negatif oran yüzdeleri (N) (True positive (TPR %) and true negative (TNR %) rate percentages of the fully (TE), standard (SE) and, dilated depthwise separable (GASE) models, with the minimum 80% true negatives condition and the minimum 99% true positive condition, for the categories)

	En az %80 Doğru Negatif			En az %99 Doğru Pozitif		
Kategori	TE	SE	GASE	TE	SE	GASE
P: Tümü	98.6	100.0	100.0	100.0	100.0	100.0
P: Ön yüz konuşma	100.0	100.0	100.0	100.0	100.0	100.0
P: Dönmüş yüz konuşma	91.7	100.0	100.0	100.0	100.0	100.0
N: Tümü	86.1	100.0	86.1	61.1	100.0	86.1
N: Kafa dönmeleri	100.0	100.0	91.7	91.7	100.0	91.7
N: Dudak yapıştırma-ısırma	87.5	100.0	75.00	62.5	100.0	75.0
N: Ağız açıp nefes alma	75.0	100.0	100.0	25.0	100.0	100.0
N: Gülümseme	100.0	100.0	100.0	75.0	100.0	100.0
N: Çene-ağız üzeri parmak	50.0	100.0	75.0	00.0	100.0	75.0
N: Yüz önü el kaydırma	75.0	100.0	75.0	50.0	100.0	75.0

5. Sonuçlar (Conclusions)

Bu çalışma, görsel konuşma sesi saptama için literatürde yaygın olarak kullanılan video kameralara verimlilik açısından daha üstün bir alternatif olarak olay kamerasının kullanımını ele almıştır. KST birçok uygulamada ön işlem olarak çalışması gerektiğinden, enerji verimliliği ve işlem karmaşıklığı kritik öneme sahiptir. Bu ihtiyaç doğrultusunda, olay kamerası sayesinde yüksek zaman çözünürlüğünde elde edilen ağız bölgesi olay yoğunluğu dizisini, evrişimsel sinir ağları ile temel alan yeni bir yöntem önerilmiştir.

Uzamsal boyut tamamen indirgendiğinden, son derece düşük hesaplama gereksinimi olan hafif modeller oluşturulmuştur. Verimliliği üst seviyelere çıkarabilen teknikleri ve tasarımları bulabilmek için farklı mimari tasarımlar karşılaştırılmıştır. KST problemini ele alan önceki bütün video kamerası veya olay kamerası çalışmalarından farklı olarak, başarım ve verimlilik birlikte kapsamlı bir şekilde değerlendirilmiştir.

Konuşma artikülasyonundan farklı yüz eylemleri durumundaki başarımları ölçebilmek için, veri kümesinde çeşitli yüz aktivitesi klipleri de dahil edilmiştir. Böylece, her bir farklı kategorideki yüz eylemleri karşısında KST'nin dayanıklılığı kategori özelinde de incelenmiştir. Konuşma dışı yüz aktivitelerinin olduğu durumlarda dahi yüksek başarımların elde edilmiş olması, hiçbir uzamsal bilgi kullanılmadan, yani sadece ayırt edici dinamik yoğunluk örüntü modellerinin öğrenilmesiyle, mükemmel seviyede saptama yapılabildiğini göstermiştir. Bu sonuç, ele alınan problemde, ağız bölgesindeki uzamsal bilginin tanıma için aslında gerekli olmadığını destekleyen bir bulgudur. Uzamsal örüntülerin modellenmesi, önemli derecede karmaşıklık ve büyük işlem yükü maliyeti getireceğinden, bunların gerekmeyecek olması verimlilik açısından çok önemli bir sonuçtur.

Zamansal alıcı alanı, sinir ağı derinliğine göre farklı hızlarda artıran aşağı-örnekleme, sabit evrişim genleştirme ve, üstel evrişim genleştirme teknikleri incelenmiştir. Deneysel sonuçlar, öncelikle, aşağı-örneklemenin hem yüksek başarım hem de yüksek verimlilik için gerekli olduğunu göstermiştir. Ancak, aşağı-örneklemeyi gerçekleştiren maksimum-havuzlama uygulamasının (SE), daha az işlem yükü ile çalışan adımlı evrişim aşağı-örneklemesinden (TE) önemli ölçüde daha yüksek başarım elde ettiği bulunmuştur. Bunun sebebi, maksimum-havuzlama ile daha kuvvetli öteleme değişmezliği elde edilmesi olabilir; çünkü, konuşma aralıkları son derece değişken olarak ortaya çıkar. Ayrıca, derinliğin çok fazla arttırılmasının ise (altı katmandan fazla) başarım ve verimliliği olumsuz etkilediği görülmüştür. Başarım-verimlilik dengesi açısından bakıldığında, aşağı-örnekleme ve maksimum havuzlama yapan altı katmanlı modelin (SE) mükemmel başarım sağladığı ancak, beş katmanlı fakat sabit oranla genleştirilmiş evrişim uygulayarak aynı zamansal alıcı alana erişen modelin (GSE), sadece %0.3'lük ufak bir başarım düşüşüyle, verimliliği iki kattan fazla artırdığı görülmüştür (1.57'den 0.75 MFLOPS'a). Verimliliği daha da artırabilmek amacıyla, evrişim çekirdeklerini kanal ekseni üzerinde ayıran derinlemesine ve farklı gruplamalı ayırma tasarımları karşılaştırılmıştır. Deneyler, verimliliği en yüksek seviyede artıran derinlemesine ayırmanın (GASE-Der), GSE'nin başarımını koruyarak işlem yükünü 0.30 MFLOPS'a indirdiğini, yani SE'nin beşte birinden daha az işlem gücüyle de KST yapılabileceğini göstermiştir. Dolaysıyla, verimliliğin kritik olduğu uygulamalarda GASE-Der modeli, SE'ye göre %0.3'lük ufak bir başarım düşüşüyle, tercih edilebilecek bir modeldir.

Bu çalışmada elde edilen sonuçlar, olay kamerası ile KST çözümünün çok verimli bir alternatif olabileceğini göstermiş olup bu alandaki başka araştırmaları motive edicidir. İleride ele alınabilecek bir konu, gerçek zamanlı uygulamalar için önemli olan gecikme süresi olabilir. Sabit bir zaman penceresi kullanılarak tespit yapıldığında, çıktı zamanı pencere merkezi olarak varsayılır, yani aynı sayıda geçmiş ve gelecek girdi örneği kullanılır. Bu durumda gecikme süresi zamansal alıcı alanla orantılıdır. Öte yandan, nedensel evrişim filtrelemesi yapılırsa, yani bütün girdiler geçmiş zaman örnekleri olursa, sıfır gecikme süresinde KST yapılabilir. Ancak bunun başarımı olumsuz etkilemesi beklenir; çünkü, konuşmanın belli bir bağlam süresi vardır. Dengeli bir çözüm, farklı oranda asimetrik zaman pencereli evrişim yapmak olabilir; yani fazla sayıda geçmiş örnek ama az sayıda gelecek örnekle çalışan evrişim modelleri tasarlanabilir. Dolayısıyla, sonraki bir çalışmada, gecikme süresini bu şekilde en aza indiren tasarımlar hedeflemek önemli bir araştırma konusu olacaktır.

Teşekkür (Acknowledgment)

Bu çalışma, Yaşar Üniversitesi Proje Değerlendirme Komisyonu (PDK) tarafından kabul edilen BAP112 no.lu ve "Nöromorfik Kamera ile Dinamik Yüz Analizi" başlıklı proje kapsamında desteklenmiştir.

Kaynaklar (References)

Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., Nayak, T., Andreopoulos, A., Garreau, G., Mendoza, M., Kusnitz, J., Debole, M., Esser, S., Delbruck, T., Flickner, M., Modha, D., 2017. A Low Power, Fully Event-Based Gesture Recognition System. CVPR2017, The IEEE/CVF Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA.

Araujo, A., Norris, W., Sim, J., 2019. Computing Receptive Fields of Convolutional Neural Networks. Distill, https://distill.pub/2019/computing-receptive-fields.

Ariav, I., Dov, D., Cohen, I., 2018. A deep architecture for audio-visual voice activity detection in the presence of transients. Signal Processing 142, 69–74.

Arriandiaga, A., Morrone, G., Pasa, L., Badino, L., Bartolozzi, C., 2021. Audio-Visual Target Speaker Enhancement on Multi-Talker Environment Using Event-Driven Cameras. ISCAS 2021, IEEE International Symposium on Circuits and Systems, Daegu, South Korea, May 22-28, 2021.

Bai, S., Kolter, J.Z., Koltun, V., 2018. Convolutional Sequence Modeling Revisited. ICLRW2018, 6th International Conference on Learning Representations - Workshop Track Proceedings, April 30 - May 3, 2018, Vancouver, BC, Canada.

Barua, S., Miyatani, Y., Veeraraghavan, A., 2016. Direct face detection and video reconstruction from event cameras. WACV2016, Winter Conference on Applications of Computer Vision, March 7-10, 2016, Lake Placid, NY, USA.

Berlincioni, L., Cultrera, L., Albisani, C., Cresti, L., Leonardo, A., Picchioni, S., Becattini, F., Del Bimbo, A., 2023. Neuromorphic Event-based Facial Expression Recognition. CVPRW2017, The IEEE/CVF Conference on Computer Vision and Pattern Recognition - Workshop Track., June, 2023, Vancouver, Canada, pp. 4108–4118.

Çubukçu, A., Kuncan, M., Kaplan, K., Ertunç, H.M., 2015. Development of a voice-controlled home automation using Zigbee module. In: 23nd Signal Processing and Communications Applications Conference (SIU). pp. 1801–1804.

Deng, Y., Chen, H., Liu, H., Li, Y., 2022. A Voxel Graph CNN for Object Classification With Event Cameras. CVPR2022, The IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, USA, June 2022.

Gallego, G., Lund, J.E.A., Mueggler, E., Rebecq, H., Delbrück, T., Scaramuzza, D., 2018. Event-Based, 6-DOF Camera Tracking from Photometric Depth Maps. IEEE Trans. Pattern Anal. Mach. Intell. 40, 2402–2412.

Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A.J., Conradt, J., Daniilidis, K., Scaramuzza, D., 2022. Event-Based Vision: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 44, 154–180.

Gehrig, D., Loquercio, A., Derpanis, K.G., Scaramuzza, D., 2019. End-to-End Learning of Representations for Asynchronous Event-Based Data, ICCV2019, The IEEE International Conference on Computer Vision, October 2019.

Ghaemmaghami, H., Dean, D., Kalantari, S., Sridharan, S., Fookes, C., 2015. Complete-linkage clustering for voice activity detection in audio and visual speech. Interspeech, Dresden, Germany, 2015.

Guy, S., Lathuilière, S., Mesejo, P., Horaud, R., 2020. Learning Visual Voice Activity Detection with an Automatically Annotated Dataset. ICPR2020, 25th International Conference on Pattern Recognition, January 10-15, 2020, Milan, Italy.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arxiv:1704.04861.

Kim, J., Hwang, I., Kim, Y.M., 2022. Ev-TTA: Test-Time Adaptation for Event-Based Object Recognition. CVPR2022, The IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, USA, June 2022.

Korkmaz, Y., Boyacı, A., 2023. Hybrid voice activity detection system based on LSTM and auditory speech features. Biomedical Signal Processing and Control 80, 104408.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. NIPS2012, Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems 2012, December 3-8, 2012, Lake Tahoe, Nevada, USA.

Lenz, G., Ieng, S.H., Benosman, R.B., 2020. Event-based Face Detection and Tracking using the Dynamics of Eye Blinks. Frontiers in Neuroscience 14, 587.

Li, J., Li, J., Zhu, L., Xiang, X., Huang, T., Tian, Y., 2022. Asynchronous Spatio-Temporal Memory Network for Continuous Event-Based Object Detection. IEEE Transactions on Image Processing 31, 2975–2987.

Li, X., Neil, D., Delbruck, T., Liu, S., 2019. Lip Reading Deep Network Exploiting Multi-Modal Spiking Visual and Auditory Sensors. ISCAS 2019, IEEE International Symposium on Circuits and Systems, May, 2019.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. CVPR2015, The IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2015, Boston, USA.

Maqueda, A.I., Loquercio, A., Gallego, G., García, N., Scaramuzza, D., 2018. Event-Based Vision Meets Deep Learning on Steering Prediction for Self-Driving Cars. CVPR2018, The IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, USA, June 2018.

Moreira, G., Graça, A., Silva, B., Martins, P., Batista, J.P., 2022. Neuromorphic Event-based Face Identity Recognition. ICPR2022, 26th International Conference on Pattern Recognition, Montreal, August 21-25, 2022, QC, Canada, pp. 922–929.

Neil, D., Pfeiffer, M., Liu, S.-C., 2016. Phased LSTM: Accelerating Recurrent Network Training for Long or Event-based Sequences. NIPS2016, Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, pp. 3889–3897.

Pan, L., Scheerlinck, C., Yu, X., Hartley, R., Liu, M., Dai, Y., 2019. Bringing a Blurry Frame Alive at High Frame-Rate With an Event Camera. CVPR2019, The IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, June 2019.

Paredes-Valles, F., de Croon, G.C.H.E., 2021. Back to Event Basics: Self-Supervised Learning of Image Reconstruction for Event Cameras via Photometric Constancy. CVPR2021, The IEEE/CVF Conference on Conference on Computer Vision and Pattern Recognition, June 2021.

Patrona, F., Iosifidis, A., Tefas, A., Nikolaidis, N., Pitas, I., 2016. Visual Voice Activity Detection in the Wild. IEEE Transactions on Multimedia 18, 967–977.

Perot, E., de Tournemire, P., Nitti, D., Masci, J., Sironi, A., 2020. Learning to Detect Objects with a 1 Megapixel Event Camera. NIPS2020, Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, December 6-12, 2020.

Rebecq, H., Ranftl, R., Koltun, V., Scaramuzza, D., 2019. Events-To-Video: Bringing Modern Computer Vision to

Event Cameras. CVPR2019, The IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, June 2019.

Rethage, D., Pons, J., Serra, X., 2018. A Wavenet for Speech Denoising. ICASSP2018, IEEE International Conference on Acoustics, Speech and Signal Processing, April 15–20, 2018 Calgary, Alberta, Canada, pp. 5069–5073.

Ryan, C., O'Sullivan, B., Elrasad, A., Cahill, A., Lemley, J., Kielty, P., Posch, C., Perot, E., 2021. Real-time face & eye tracking and blink detection using event cameras. Neural Networks 141, 87–97.

Savran, A., Tavarone, R., Higy, B., Badino, L., Bartolozzi, C., 2018. Energy and Computation Efficient Audio-Visual Voice Activity Detection Driven by Event-Cameras. FG2018, 13th IEEE International Conference on Automatic Face & Gesture Recognition, May 15-19 2018, Xi'an, China.

Savran, A., Bartolozzi, C., 2020. Face Pose Alignment with Event Cameras. Special Issue: Sensor Systems for Gesture Recognition, Vol. 20, Issue 24, Article 7079.

Savran, A., 2023. Multi-timescale boosting for efficient and improved event camera face pose alignment. Computer Vision and Image Understanding, Vol. 236, 103817.

Savran, A., 2023a. Fully Convolutional Event-camera Voice Activity Detection Based on Event Intensity. ASYU2023, IEEE Innovations in Intelligent Systems and Applications Conference, October, 2023, Sivas, Türkiye.

Savran, A., 2023b. Comparison of Timing Strategies for Face Pose Alignment with Event Camera. In: 8th International Conference on Computer Science and Engineering (UBMK). pp. 97–101.

Schaefer, S., Gehrig, D., Scaramuzza, D., 2022. AEGNN: Asynchronous Event-Based Graph Neural Networks. CVPR2022, The IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, USA, June 2022.

Shahid, M., Beyan, C., Murino, V., 2021. S-VVAD: Visual Voice Activity Detection by Motion Segmentation. WACV2021, Winter Conference on Applications of Computer Vision, January 3-8, 2021, Waikoloa, HI, USA, pp. 2331-2340

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. CVPR2015, The IEEE/CVF Conference on Conference on Computer Vision and Pattern Recognition, June 2015, Boston, USA.

Sharma, R., Somandepalli, K., Narayanan, S.S., 2019. Toward Visual Voice Activity Detection for Unconstrained Videos. ICIP2019, International Conference on Image Processing, September 22-25, 2019, Taipei, Taiwan.

Tan, G., Wang, Y., Han, H., Cao, Y., Wu, F., Zha, Z.-J., 2022. Multi-Grained Spatio-Temporal Features Perceived Network for Event-Based Lip-Reading. CVPR2022, The IEEE/CVF Conference on Conference on Computer Vision and Pattern Recognition, New Orleans, USA, June 2022.

Tulyakov, S., Bochicchio, A., Gehrig, D., Georgoulis, S., Li, Y., Scaramuzza, D., 2022. Time Lens++: Event-Based Frame Interpolation With Parametric Non-Linear Flow and Multi-Scale Fusion. CVPR2022, The IEEE Conference on Conference on Computer Vision and Pattern Recognition, New Orleans, USA, June 2022.

Wang, D., Xiao, X., Kanda, N., Yoshioka, T., Wu, J., 2023. Target Speaker Voice Activity Detection with Transformers and Its Integration with End-To-End Neural Diarization. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Wang, Y., Du, B., Shen, Y., Wu, K., Zhao, G., Sun, J., Wen, H., 2019. EV-Gait: Event-Based Robust Gait Recognition Using Dynamic Vision Sensors. The IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, June 2019.

Wang, Y., Zhang, X., Shen, Y., Du, B., Zhao, G., Cui, L., Wen, H., 2022. Event-Stream Representation for Human Gaits Identification Using Deep Neural Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 44, 3436–3449.

Wrench, A., 2006. MOCHA-TIMIT, www.cstr.ed.ac.uk/research/projects/artic/mocha.html.

Yu, F., Koltun, V., 2016. Multi-Scale Context Aggregation by Dilated Convolutions. 4th International Conference on Learning Representations, ICLR, San Juan, Puerto Rico, May 2016.

Zhang, X.-L., Wang, D., 2016. Boosting Contextual Information for Deep Neural Network Based Voice Activity Detection. IEEE/ACM Transactions on Audio, Speech, and Language Processing 24, 252–264.

Zhang, J., Dong, B., Zhang, H., Ding, J., Heide, F., Yin, B., Yang, X., 2022. Spiking Transformers for Event-Based Single Object Tracking. CVPR2022, The IEEE Conference on Conference on Computer Vision and Pattern Recognition, New Orleans, USA, June 2022.

Zhu, L., Wang, X., Chang, Y., Li, J., Huang, T., Tian, Y., 2022. Event-Based Video Reconstruction via Potential-Assisted Spiking Neural Network. CVPR2022, The IEEE Conference on Conference on Computer Vision and Pattern Recognition, New Orleans, USA, June 2022.

Research Article
Journal of Intelligent Systems: Theory and Applications 7(2) (2024) 116-128
DOI: 10.38016/jista.1468153

A Hybrid Metaheuristic based Feature Selection Framework for In-silico Mutagenicity Prediction

Özlem Yılmaz[1], Enis Gümüştaş[2], Ayça Çakmak Pehlivanlı[3*]

[1] Faculty of Science and Letters, Mathematics Department, Mimar Sinan Fine Arts University, İstanbul, Türkiye

[2, 3] Faculty of Science and Letters, Statistics Department, Mimar Sinan Fine Arts University, İstanbul, Türkiye

ozlem.yilmaz@msgsu.edu.tr, 20203107001@ogr.msgsu.edu.tr, ayca.pehlivanli@msgsu.edu.tr

Abstract

Mutagenicity is both a toxic risk to humans and an indicator of carcinogenicity. Hence, estimating mutagenicity in the early stages of drug design is crucial to minimize last-stage failures and withdrawals in drug discovery. Recently, in-silico methods have started to play critical and essential roles in the drug development process because they are low cost and low effort procedures. This study aims to predict mutagenicity of chemicals using in-silico methods. To achieve this goal, a two-phased flexible framework was proposed: 1) searching the effective and representative descriptors subset with Butterfly Optimization Algorithm (BOA) and Particle Swarm Optimization and 2) predicting mutagenicity of chemicals by the selected descriptor using gradient boosted tree-based ensemble methods. The study used two datasets: one including 8167 compounds for descriptor selection and modelling, and another containing 716 external compounds to validate the efficacy of our models. The datasets comprise 162 descriptors calculated using PaDEL. The results of both the cross-validation and the external data showed that descriptors reduced by nearly one-third by BOA (51 descriptors) yielded similar or slightly better predictive results than results obtained with the entire data set. The accuracy range attained by the proposed approach using BOA is approximately 91.9% to 97.91% for the external set and 83.35% to 86.47% for the test set. This research contributes that using optimization techniques for improving early drug design and minimizing risks in drug discovery can be considered as a valuable insights and advances in the field of drug toxicity prediction, based on the findings.

Keywords: Machine Learning, Feature Selection, Metaheuristics, Gradient Boosting Algorithms, Mutagenicity Prediction, In-Silico Modelling

In-silico Mutajenite Tahmini için Hibrit Metasezgisel Tabanlı Özellik Seçimi Çerçevesi

Öz

Mutajenite hem insanlar için toksik bir risk hem de kanserojenitenin bir göstergesidir. Bu nedenle, ilaç tasarımının erken aşamalarında mutajenitenin tahmin edilmesi, ilaç keşfinde son aşama başarısızlıklarını ve geri çekilmeleri en aza indirmek için çok önemlidir. Son zamanlarda, in-silico yöntemler, düşük maliyetli ve az çaba gerektiren prosedürler olmaları nedeniyle ilaç geliştirme sürecinde kritik ve önemli roller oynamaya başlamıştır. Bu çalışma, in-silico yöntemler kullanarak kimyasalların mutajenitesini tahmin etmeyi amaçlamaktadır. Bu amaca ulaşmak için iki aşamalı esnek bir çerçeve önerilmiştir: 1) Kelebek Optimizasyon Algoritması (BOA) ve Parçacık Sürü Optimizasyonu ile etkili ve temsili değişken alt kümesinin aranması ve 2) gradyan destekli ağaç tabanlı topluluk yöntemleri kullanılarak seçilen değişkenlere göre kimyasalların mutajenitesinin tahmin edilmesi. Çalışmada iki veri kümesi kullanılmıştır: biri değişken seçimi ve modelleme için 8167 bileşik, diğeri ise modellerimizin etkinliğini doğrulamak için 716 harici bileşik içermektedir. Veri kümeleri PaDEL kullanılarak hesaplanan 162 değişkeni içermektedir. Hem çapraz doğrulama hem de harici verilerin sonuçları, BOA ile neredeyse üçte bir oranında azaltılan değişkenlerin (51 adet), tüm veri setiyle elde edilen sonuçlara benzer veya biraz daha iyi tahmin sonuçları verdiğini göstermiştir. BOA kullanılarak önerilen yaklaşımla elde edilen doğruluk aralığı harici set için yaklaşık %91,9 ila %97,91 ve test seti için %83,35 ila %86,47'dir. Bu araştırma, bulgulara dayanarak, erken ilaç tasarımını iyileştirmek ve ilaç keşfindeki riskleri en aza indirmek için optimizasyon tekniklerinin kullanılmasının, ilaç toksisitesi tahmini alanında değerli bir içgörü ve ilerleme olarak kabul edilebileceğine katkıda bulunmaktadır.

Anahtar Kelimeler: Makine Öğrenmesi, Özellik Seçimi, Metasezgisel, Gradyan Boosting Algoritmaları, Mutajenite Tahmini, In-Silico Modelleme

* Corresponding Author
E-Mail: ayca.pehlivanli@msgsu.edu.tr

Recieved : 14 Apr 2024
Revision : 16 Aug 2024
Accepted : 20 Aug 2024

1. Introduction

Mutagenicity can be defined as the capacity of a compound to cause permanent mutations in the DNA sequence (Bakhtyari et al. 2013). It could lead to a toxic risk to humans. Moreover, it is the indicator of carcinogenicity which means healthy cell transforms themselves into cancer cells. Assessment of mutagenicity at the early stages of the drug approval process is crucial to swift eliminate such compounds from the drug development pipeline (Raghavan 2005). Among the toxicity tests, the in-vitro Ames test has become a standard for assessing mutagenicity (Zhang 2017). However, these in-vitro experiments are highly expensive, laborious, and time-consuming. On the other hand, both in-vivo experiments and in-vitro experiments have begun to give up their places to statistical and computational methods and tools developed in the computer environment without requiring laboratory experiments. Therefore, to prevent drug failure and withdrawal due to late-stage mutagenicity, it is necessary to predict mutagenicity by developing in-silico methods. Computational methods used for in-silico approaches can be grouped basically as expert rule-based systems, also referred to as structural alerts, and statistics-based models, known as quantitative structure-activity relationship tools (Bakhtyari et al. 2013; Honma 2019; Hansch 1980). Difficulties in explainability and interpretability, which are the main drawbacks of computational approaches, are almost non-existent in expert systems. Therefore, expert systems are widely used because they provide comprehensive outputs that can be understood, questioned and judged by the user. Despite this transparency, their prediction success is lower than that of statistical approaches (Wichard 2017). It is worth noting that expert systems for in-silico research are available both commercially and open access (Honma 2019; Çakmak Pehlivanlı and Çakmak 2022).

A well-designed in silico approach can yield several benefits, including the ability to plan studies with fewer animals, identify the concentration that will be used in advance, save time and money, and guide whether the information obtained about the molecule should proceed to laboratory experiments (Toropov et al. 2014).

Metaheuristic optimization algorithms are ideally suited for efficiently exploring the complex and high-dimensional feature spaces encountered in feature selection problems due to their stochastic, adaptable, and global search characteristics.

To the best of our knowledge, no studies address the estimation of drug toxicity, particularly mutagenicity, except for the limited number of studies in which metaheuristic optimization algorithms have been applied to drug discovery (Houssein et al. 2020; Algamal et al. 2020; Subaş and Çakmak Pehlivanlı 2020). The main contributions and scope of this paper are summarized as follows in order to fill this gap;

- We propose a flexible approach that hybridized metaheuristic algorithms with several machine learning algorithms to select descriptors and compare the classification models that promise the best prediction results of mutagenicity,
- We show that metaheuristic algorithms and machine learning algorithms can work together in in-silico studies such as drug toxicity prediction,
- We conclude whether metaheuristic approaches are suitable for searching the descriptor space and enhancing mutagenicity classification based on the chosen descriptors.

It should be noted that our motivation is not only the success of the optimization part but also mostly obtaining higher accuracy with fewer descriptors. To address these aforementioned aims, we introduce a flexible approach by hybridizing metaheuristic and machine learning algorithms on mutagenicity.

The rest of this paper is organized as follows. After introducing the related work of this paper in Section 2, dataset and the proposed model are described in Section 3. Experimental results, discussions and conclusions are presented in Sections 4, 5 and 6 respectively.

2. Related works

Early studies on systems based on rules and expert knowledge gained speed, especially at the end of the ’90s. The relationships between chemical structures and observed toxic effects and outcomes were examined, and various software was presented comparatively (Greene et al. 1999). This study was followed by several in-silico studies (White et al. 2003; Cariello et al. 2002).

In the early 2000s, modelling based on statistical learning algorithms was commonly used to predict mutagenicity in in-silico studies. Zheng et al. developed a mutagenic probability model with support vector machines (SVM) for the mutagenicity prediction and achieved better performance than the TOPKAT, a tool based on rules and expert knowledge (Zheng 2006). Liao et al. applied a combination of recursive partitioning (RP) and SVM on different data sets to predict mutagenic toxicity and achieved between 80.2% and 87.3% performances with two models (Liao et al. 2007). In order to improve in-silico methodologies used to predict mutagenicity in the first decade of the 2000s, Mazzatorta et al. proposed a novel system named robust hybrid classifier (RHC) by combining a fragment-based structure activity relationship (SAR) model and AI-based approaches on Bursi mutagenicity data set. The performance of the proposed methods was tested with external test data and obtained 85% both in sensitivity and specificity (Mazzatorta et al. 2007; Kazius et al. 2005). In order to build a public Ames mutagenicity data set, Hansen et al. constructed a data set that comprised about 6500 compounds, in the format of SMILES (simplified molecular input line entry specification) and

SDF (structure data format), with biological activity (Hansen 2009). In the same study, this benchmark data set was used to compare commercial tools (DEREK, Pipeline Pilot, and MultiCase) based on expert knowledge with machine learning algorithms (SVM, random forest (RF), k-nearest neighbour (KNN), and Gaussian process (GP)). As a result of this study, while the best performance was obtained by SVM with 0.86 AUC, DEREK yielded the lowest sensitivity and specificity (Hansen 2009).

Since toxicity is one of the most critical issues that cause late-stage drug failure or withdrawal, the in-silico studies in the prediction of mutagenicity gained speed in the last decade. In most of these studies, machine learning and statistical learning-based methods such as SVM, RF, artificial neural networks (ANN), KNN, genetic algorithms, radial basis function (RBF), partial least squares (PLS), naïve Bayes methods were preferred (Sharma et al. 2011; Webb et al. 2014b; Xu et al. 2012). Since the experimental screening of chemical compounds for biological activity is time consuming and expensive, Seal et al. applied supervised learning approaches on two different data sets to generate an alternative predictive model. As a result of the study, the RF algorithm achieved the best performance with 89.27% with a new mutagenicity data set comprising two well-known data (Seal et al. 2012). Webb et al. published a study emphasizing that interpretability of Ames mutagenicity prediction is more important than successful performance to interpret the model. They tried to extract the pattern of biological activity through the descriptors importance (Webb et al. 2014a). Zhang aimed to investigate the prediction of agents as mutagens and non-mutagens using a naive Bayes classifier in several studies. In addition to this purpose, they focused on identifying the most informative molecular descriptors related to mutagenicity. Although the prediction performance was similar to previous studies, their model identified four simple molecular descriptors (apol, number of H donors, number of rings, and Wiener) related to mutagenicity (Zhang 2017, 2015, 2016). Another research group has provided machine learning-based models for toxicity prediction of approximately 1500 diverse chemical compounds in various species. It has been reported that 70% of compounds were classified correctly based on the random forest algorithm and listed the physicochemical descriptors based on their importance (Moorthy et al. 2017).

Several kinds of research have recently been published on the prediction of toxicity and in-silico drug discovery with new approaches such as deep learning and ensemble methods such as XGBoost (Fan 2018; Rifaioğlu et al. 2019; Ji et al. 2019). Most recent studies in this area generally examine the current impact of AI studies on drug toxicity, potential challenges and future perspectives and potential (Tran et al. 2023; Zhang et al. 2019; Chu et al. 2021). In their 2023 review, Tran et al. provide an overview of recent AI driven advances in drug toxicity prediction, including machine learning and deep learning techniques on various toxicity traits (Tran et al. 2023). Zhang et al. conducted a study on chemical toxicity prediction with LightGBM, a machine learning algorithm, using Tox21 and Mutagenicity databases (Ji et al. 2019). Similarly, Chu et al. tried to present robust in silico models accurately estimate a compound's mutagenicity before synthesis to get around the limitations (costly, time consuming) of the Ames test (Chu et al. 2021).

Providing the most related descriptors that effectively classify mutagenic and non-mutagenic compounds also emerges as another important research area. Feature selection which can be conducted either based on the wrapper or filter approach, is considered a preprocessing for machine learning algorithms. It is generally hard to obtain the best feature subset set using traditional approaches. Therefore, metaheuristic approaches can be another alternative in order to select optimal subsets. In 2020, Houssein et al. built a novel hybrid Harris Hawks optimization (HHO) and SVM in drug discovery. As they reported, this was the first time HHO had been applied in the field of drug design (Houssein et al. 2020). Similarly, Algamal et al. developed the pigeon optimization algorithm with a new time varying transfer function to select the features most relevant to high dimensional QSAR / QSPR classification modeling (Algamal et al. 2020).

3. Material and Methods

3.1. Data preparation

This study used a combination of two popular data sets, namely The Benchmark and Bursi Mutagenicity data sets. The Benchmark data set consists of 6512 compounds, and the Bursi data set has 4337 compounds. These data sets were collected by Hansen et al. and Kazius et al., respectively (Kazius et al. 2005; Hansen 2009). According to Ames results, each compound in the data sets was given its canonical SMILES format and corresponding label, indicating whether it was mutagen or non-mutagen. In addition to this training data set, an external validation set has been included to study to make a fair measurement of the proposed approach. The external data set consisted of canonical SMILES format of 731 compounds was collected by Xu et al. (Xu et al. 2012). After removing duplicate compounds based on their SMILES format, 8167 unique compounds in the training set and 716 unique compounds in the external validation set were left. All the descriptors of molecules were calculated by PaDEL-Descriptor software (Yap 2010). Among 1444 1D and 2D physicochemical descriptors, i.e., properties, 225 descriptors were chosen based on several studies (Xu et al. 2012; Fan 2018; Gupta and Rana 2019; Guan et al. 2018). During the preprocessing and selection process, only the training data set was used. The limited number of missing values was imputed by using mean. Correlated descriptors given in the correlation matrix across all pairs of descriptors with 0.95 or higher correlations were assumed to be redundant and removed from the data set. Finally, the entire data set used in this study consisted of 162 descriptors, and the details were presented in Table 1.

3.2. Butterfly optimization algorithm

BOA, proposed by Arora et al., is a metaheuristic algorithm that models the food foraging behavior of butterflies in nature (Arora and Singh 2019). Through chemoreceptors scattered on their bodies, butterflies can separate different fragrances of food (flowers), sense (smell) their intensities, and perform foraging movements (Tubishat et al. 2020). During their movements, butterflies can produce fragrance with an intensity that is directly proportional to their fitness. Butterflies communicate with each other by the fragrance they emit. BOA is a global optimization method based on the communication behaviors of butterflies. The intensity of the fragrance a butterfly emits is as much as other butterflies can feel it. The most crucial feature of BOA that differs from different metaheuristic algorithms is that the intensity of fragrance felt by the butterfly is unique. The most critical part of BOA algorithm is how the fragrance is calculated based on concepts of sensing and processing the modality like the smell, sound, temperature, etc. As reported in the original study of Arora et al., modality is fragrance in BOA Arora and Singh (2019). Three terms should be clearly explained for this; sensory modality (c), stimulus intensity (I), and power exponent (a). The formulation of the perceived fragrance intensity for each butterfly is given in Eq.(1) based on Steven's Law of Power (Arora and Singh 2019; Stevens 1986).

$$f = cI^{a} \tag{1}$$

where *f* is the emitted magnitude of the fragrance, i.e., how intensively other butterflies emit the fragrances within the search space, *c* is a proportionality constant taken as the sensory modality taken in the range [0, 1], *I* is the stimulus magnitude of the perceived fragrance by butterfly, and *a* is the power exponent characterizing the degree of absorption of sensory modality with its values over the range [0, 1]. Since *a* and *c* directly affect the convergence speed of the BOA algorithm, it is a crucial point to choose suitable values for both *c* and *a*. This can be expressed as, at the extreme points of the range, the fragrance emitted by the butterfly, if *a* = 0 it is not perceived by other butterflies if *a* = 1, it is perceived by other butterflies at the same intensity.

Butterflies share information with each other about their positions according to the fragrance intensity they produce. Thus, the butterflies change their positions towards the best butterfly closest to the food with the optimum fragrance intensity in the search space. This movement of butterflies is called global search and determined as in Eq. (2).

$$x_i^{t+1} = x_i^t + (r^2 \times g^* - x_i^t) \times f_i \tag{2}$$

where the solution vector x_i^t is the position of ith butterfly in movement t, g^* is the current fittest position decided among the available positions at the current movement of all butterflies, sensed fragrance magnitude of the butterfly is symbolized by f_i and r is a uniform random number in the range of [0, 1].

On the other hand, when butterflies cannot detect the fragrance of different butterflies in the search space, they move randomly. This movement of butterflies is called local search and formulated as in Eq. (3).

$$x_i^{t+1} = x_i^t + \left(r^2 \times x_j^t - x_k^t\right) \times f_i \tag{3}$$

where the solution vector x_j^t and x_k^t are the position of jth and kth butterfly in movement t. If x_j^t and x_k^t are located in the same neighborhood, and r is a random number in the range of [0, 1], then Eq. (3) turns out to be a local random stride. In order to control switching between global search and local search space in BOA, the switching probability (p) parameter is utilized.

3.3. Particle swarm optimization algorithm

First introduced by Kennedy et al., PSO is one of the metaheuristic search algorithms inspired by the bird's swarm's social behaviour (Kennedy and Eberhart 1995). PSO is a population-based algorithm that consists of the particles, i.e., a possible set of solutions. These particles move through in the multidimensional search space in order to find the best solution. While their movement, they have a memory in keeping track of their previous best position, namely best solution. Besides concerning their own best solutions, they considered the best solution of the swarm as well (Mirjalili and Lewis 2013). There are two types of particle positions, namely local (personal) best and global best.

Table 1. Distribution of the mutagens and non-mutagens in training and external validation set

Data Sets	Mutagens	Non-Mutagens	Total
Training data set (Kazius et al. 2005; Hansen 2009)	4524	3643	8167
External validation set (Xu et al. 2012)	599	117	716
Total	5123	3760	8883

Each particle owns certain information in order to update its position; the current position, the current velocity, distance to the local best solution (p), and distance to the global best solution (g^*). The mathematical definition of the PSO model consists of both the velocity of the ith particle at iteration $t + 1$ and also the new position of the ith particle given in Eq. (4) and Eq. (5), respectively.

$$v_i^{t+1} = wv_i^t + c_1.r(p_i - x_i^t) + c_2.r(g^* - x_i^t) \tag{4}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{5}$$

where v_i^t is the velocity of the ith particle at iteration t, c_1 and c_2 are acceleration constants, r is a uniform random number in the range of [0, 1], w is the inertia weighting function predefined by the user, x_i^t represents the current position of ith particle at iteration t, p_i is the best solution that obtained previously by ith particle, and global best solution $g*$ is the best position of all particles, i.e., swarm. Once v_i^{t+1} (the velocity of the ith particle at iteration $t + 1$) is obtained by Eq. (4), the position of the ith particle is updated by Eq. (5).

The general idea of Eq. (4) can be explained as the combination of exploration ability wv_i^t, individual thinking ($c_1 \times r \times (p_i - x_i^t)$) and collaboration of particles ($c_2 \times r \times (g^* - x_i^t)$). Initially, each particle placed in the search space has a random position, velocity, and fitness value calculated by the fitness function. At each iteration, the velocity and the position of particles are updated until the stopping criterion is met (Mirjalili and Lewis 2013).

3.4. Machine learning based models

Machine learning as a branch of artificial intelligence seeks to build analytical computational models by learning automatically from data and improving with experience (Mitchell 1997). In this work, several machine learning methods have been used both for feature selection and prediction phases. Support vector machine (SVM) as a binary learning machine is based on statistical learning theory introduced by Vapnik (Vapnik 1995). SVM aims to construct a decision hyperplane that the margin of separation between a set of objects of different classes is maximized (Haykin 2011). K-nearest neighbour (KNN), the most basic instance-based method, was designed to approximate real-valued or discrete valued target functions with instances consisting of the k closest training examples in the training data (Mitchell 1997; Cover and Hart 1967). Logistic regression (LR), known as the discriminative classifier, is one of the baseline machine learning algorithms used widely for binary and multinomial classifications. Random forest (RF), proposed by Breiman, is one of the commonly used ensemble algorithms. Basically, it combines the results of the tree predictors applying on the random subsamples from a standard data set. The strength of the individual trees in the forest and their correlation affects the generalization error of a forest (Breiman 2001). Randomness, the most crucial property of RF, can be defined as a combination of bagging and random subspace methods (Ho 1998). Extremely randomized trees (ExTrees) and RF can be similar ensemble algorithms that follow almost identical procedures. The main differences are in the subsampling approach and the split points selections. While RF obtains subsamples with replacement, ExTrees uses the original sample. The cut points are decided randomly in ExTrees, whereas RF selects the optimum cut point (Geurts et al. 2006). Extreme gradient boosting (XGBoost) is an ensemble of decision tree models utilized based on the principle of gradient boosting machines (Chen and Guestrin 2016). Although training based on the gradient boosting principle can be diffcult, it can achieve a lower model bias than the RF. XGBoost follows the idea to correct the previous mistakes done by the model and propagate the experience to the next step for improving the performance. Light gradient boosting (LightGBM) like XGBoost is also a supervised learning method based on the gradient boosting framework. The main difference between them is faster training speed, especially on a large data set. LightGBM is a histogram-based algorithm with low memory usage since it transforms numerical values to discrete bins (Ke et al. 2017).

3.5. Model evaluation

The data sets used in this study have binary classes as mutagen and non-mutagen. Since our main focus is to identify mutagen compounds, mutagen labelled class is assumed as 1 which indicates positive class, and non-mutagen labelled class is accepted as 0, i.e. negative class. True positive (TP) and true negative (TN) results show that the compound is correctly predicted to be mutagenic and non-mutagenic, respectively. On the other hand, false positive (FP) and false negative (FN) results indicate that the compound has been incorrectly predicted to be positive and negative, respectively (John et al., 2023). In this study, several statistics had been calculated based on the confusion matrix to measure the models' performance. All the experiments were evaluated in terms of sensitivity (recall), specificity, F-measure, and accuracy. In addition to these measurements, AUC and probability excess had been preferred to compare models. The reason to choose the probability excess is that relative class frequencies, i.e., imbalanced class distribution, do not affect the probability excess, whereas accuracy (success rate) is affected by imbalance class distribution (Yang et al. 2005).

3.6. Proposed approach

Since each metaheuristic optimization algorithm follows different search strategies, each of them may propose different subsets of features for a given dataset. The proposed approach consists of feature selection and

Algorithm 1. Hybrid Feature Selection and Prediction

```
Inputs

$n$: number of features
$OP_i$ : Optimization algoritm, $i=1,\dots,nbOP$, $nbOP$: number of optimization algorithms
$CL_j$ : Classification algorithm, $j=1,\dots,nbCL$, $nbCL$: number of classification algorithms
$FS_j^i$: Feature Subset selected with with $OP_i$ and $CL_j$
$error$ : error obtained by classifier
$d$: number of selected features
$FFS^i$ : Final Feature Subset selected with with $OP_i$ and $CL_j$
$w_1$ = importance of the classification error ($w_1$ = 1- $w_2$)
$w_2$ = importance of the number of selected features

Feature Selection Phase

for each $OP_i$; $i=1$ to $nbOP$
    for each $CL_j$ ; $j=1$ to $nbCL$
        perform optimization algorithm $OP_i$ to get the best subset of features
            calculate fitness value by objective function
                calculate classification error by $CL_j$
                evaluate the candidate subset and get $d$
                $fitness \leftarrow w_1$ x $error$ + $w_2$ x ($d/n$)
        save the best subset of features found during the $OP_i$ process, as $FS_j^i$
    end for
    determine the final feature set by majority voting strategy by selecting the most seen features among $FS_j^i$
    and save as $FFS^i$
end for

Prediction Phase

Evaluate the performance of the classification algorithms on dataset with $FFS^i$ (Final Feature Subset obtained from
Feature Selection Phase)
```

mutagenicity prediction phases described in the pseudo code seen in Algorithm 1. The feature selection phase provided in Algorithm 1 basically selects the final significant and representative feature subsets by consensus of several hybridization of optimization algorithms with classifiers. To obtain this outcome, different machine learning algorithms had been chosen as part of the fitness function, which is essential for the optimization algorithm. The fitness function allows to determine the distance of each unit (particle, butterfly, etc) to the best solution based on the nature of the chosen optimization algorithm. The units receive a fitness value by sending their position values to the fitness function. In this frame, the optimization had a two-fold aim; obtaining the lowest error with the minimum number of features (Arora and Singh 2019). The solution for this multi-objective problem had been given as a fitness function in Eq. (6)

$$fistness = w_1 \times error(classifier) + w_2 \times \frac{d}{n} \quad (6)$$

where $error(classifier)$ is the classification error rate of the $classifier$, d is the number of selected descriptors, and n represents the number of descriptors in the original data set. The importance of accuracy and the number of selected features were weighted by w_1 and w_2, respectively and chosen as $w_2 = 1 - w_1$. Thus, a balance was provided between classification accuracy and subset length utilizing the fitness function.

Once relevant and informative descriptors from each hybrid of optimization and classifier, the final subset is determined based on the majority voting strategy by selecting the most seen features obtained from each hybrid. Several machine learning algorithms were applied on data sets consisting of descriptor subsets obtained from the feature selection phase in the mutagenicity prediction phase. In order to validate the proposed model, besides cross-validation, an external data set was also used. It should be noted that the most promising property of the proposed approach is the flexibility. It can be used either the same optimization algorithms with different classifiers or different optimization algorithms with other classifiers.

4. Experimental Results

As stated earlier, the main purpose of this study is to select the most informative descriptor subset for the early prediction of the mutagenicity by following the flow given in Algorithm 1. In order to explore the performance and effectiveness of the proposed approach, BOA and PSO were hybridized with KNN, SVM, and LR respectively for the feature selection phase, and DT, ExTree, LightGBM, RF, and XGBoost were involved for the early prediction of mutagenicity with the features obtained from feature selection phase. All the experiments were conducted by using both PaDEL data set and the external data set explained before.

Normalization was applied to the data sets before the feature selection process and the models fitted with the selected variables. In order to avoid the effect of the parameters, no data set-specific parameter optimization was performed. The parameters of the classification algorithms preferred in this study were predetermined and preferred as the same for all data sets.

The optimization part of the proposed hybrid framework was operated with classifiers using the fitness function. The hybrid framework was run with different parameter combinations using KNN, SVM, and LR for BOA and PSO, respectively. To ensure both reproducibility and diversity, combinations were run using different seeds randomly generated by a seed function for each combination. Accordingly, feature selection was made for BOA and PSO using KNN with different parameter sets including different K values, different distance metrics – Manhattan and Euclidean; SVM with different parameter sets including a different number of C coefficients, different maximum iteration numbers, and LR with different parameter sets including C, different iteration numbers.

Table 2 outlines the parameter setting for BSO and PSO. The parameter values were decided based on the outcomes of the preliminary runs. Besides these parameters, different number of population size and number of iterations were obtained applied in trial-and error manner. Considering huge number of computational time and the obtained results, population size and number of iterations were chosen as 15 and 50, respectively.

All simulations were carried out in a cloud environment on Intel(R) Xeon(R) CPU @ 2.00GHz, Linux operating system, and 16 GB RAM.

Table 2 Parameter setting for optimization algorithms.

Methods	Parameters	Values
PSO	Search domain	[0, 1]
	Interia w	0.9
	Acceleration constants [c_1, c_2]	[2, 2]
BOA	Search domain	[0, 1]
	Sensory modality (c)	0.01
	Power exponent (a)	Increased from 0.1 to 0.3 with iterations
	Switching probability (p)	0.8

The reason for preferring the cloud environment is the high resource requirement and time cost. On the other hand, mutagenicity prediction part was implemented using Python 3.8. All predictive models were performed on an AMD Ryzen 5 2600X @ 3.6GHz, Windows 10, and 16GB RAM computer.

Before feature selection, the PaDEL data set was partitioned into 80% training and 20% test set, and experiments for feature selection were conducted using only the training set. In the modelling phase, 5-fold cross validation was preferred. Different iteration numbers were tried with varying numbers of particles for BOA and PSO. As a result of this selection, about 54 trials of 3 models were conducted for both methods, and feature selection outputs were obtained. With the intention of evaluating if optimization algorithms work with mutagenetic datasets, a comprehensive statistical analysis of the best, worst, mean and standard deviations of the fitness scores, average number of features and computational time were provided in Table 3. It can be observed based on the statistical fitness measurements given in Table 3, while BOA-SVM has better fitness measures than PSO-SVM, PSO-KNN and PSO-LR are slightly better than the results of BOA-KNN and BOA-LR. Conversely, the models constructed with BOA outperformed PSO in terms of the average number of selected features and the average computational time.

Table 3 Statistical analysis obtained by the hybridized algorithms based on mean, standard deviation, best and worst of the fitness scores, average number of features and computational times

Algorithm	Mean ± SD	Best	Worst	Avg. Number of Features	Avg. Computational Time ± SD
BOA-KNN	0.2366± 0.0129	0.22616	0.26694	41.17	225.8189±97.04
PSO-KNN	0.2330± 0.0086	0.22463	0.26236	53.58	357.7957±76.22
BOA-LR	0.2755± 0.0088	0.26663	0.29611	40.83	907.8523±895.52
PSO-LR	0.2671± 0.0085	0.25900	0.29444	57.08	1490.8023±1102.32
BOA-SVM	0.2882± 0.0132	0.27664	0.31983	40.28	368.8617±279.68
PSO-SVM	0.2965± 0.0071	0.29209	0.32129	68.00	627.1484±356.65

SD: standard deviation, Avg: average

Table 4 Comparison of different classification methods with all features sets and feature subsets obtained by PSO and BOA for PaDEL Test Data

Classification Method	data sets (#of Features)	F1-score	Acc	Precision	Recall	Spec	ProbEx	AUC
DT	Baseline (162)	85.25	83.41	86.52	84.01	79.56	63.57	83.04
	PSO (87)	86.00	84.46	86.19	85.81	82.30	68.11	84.25
	BOA (51)	85.57	83.78	86.85	84.33	79.97	64.31	83.41
ExTrees	Baseline (162)	87.97	86.47	89.28	86.70	82.99	69.69	86.14
	PSO (87)	87.90	86.41	89.06	86.76	83.13	69.89	86.09
	BOA (51)	87.32	85.68	89.06	85.65	81.48	67.14	85.27
LightGBM	Baseline (162)	86.76	84.88	89.39	84.27	79.29	63.56	84.34
	PSO (87)	86.45	84.64	88.51	84.49	79.84	64.33	84.17
	BOA (51)	85.34	83.35	87.51	83.28	78.19	61.47	82.85
RF	Baseline (162)	87.71	86.23	88.73	86.72	83.13	69.84	85.93
	PSO (87)	87.40	85.92	88.18	86.64	83.13	69.77	85.65
	BOA (51)	87.89	86.47	88.62	87.17	83.81	70.99	86.22
XGBoost	Baseline (162)	87.78	86.17	89.72	85.93	81.76	67.68	85.74
	PSO (87)	87.86	86.41	88.73	87.00	83.54	70.54	86.13
	BOA (51)	87.79	86.29	88.95	86.65	82.99	69.64	85.97
Overall	Baseline (162)	87.09	85.43	88.728	85.53	81.35	66.87	85.04
	PSO (87)	87.12	85.57	88.134	86.14	82.39	68.53	85.26
	BOA (51)	86.78	85.11	88.198	85.42	81.29	66.71	84.75

DT: Decision Tree, ExTrees: Extra Trees, RF: Random Forest Acc: Accuracy, Spec: Specificity, ProbEx: Probability Excess, AUC: Area Under Curve

The variables selected from each experiment were combined, and the most repetitive (majority voting) features were chosen uniquely. By following these approaches for BOA and PSO separately, the 87 most repetitive variables among the variables selected for PSO and the first 51 most repetitive variables for BOA were chosen for the final feature subset due to the feature selection phase.

In the second phase of the study, models were fitted with PaDEL data set using treebased methods (DT, RF, XGBoost, ExTree, and LightGBM) with selected features, and predictions were obtained for both test set and External data set. In Tables 4-5, one can compare the results obtained by using a data set with full features named Baseline with 162 features and the data sets which were reduced by BOA and PSO involved in the proposed feature selection approach with 51 features and 87 features respectively in terms of F1-score, Accuracy (Acc), Precision, Recall, Specificity (Spec), Probability Excess (ProbEx) and Area Under Curve (AUC). All experiments were conducted by 5-fold cross validation.

The results presented in Table 4 were obtained using PaDEL test data reserved for testing at the beginning of the experiments. It can be analysed that although ExTrees got the highest F1 score with 162 features, there is no significant difference between the reduced data sets by using the proposed feature selection scheme. It is worth noting that results obtained with almost a third of the data set yielded similar or slightly better prediction results than the results obtained with the entire data set. It can be observed by analysing the results of BOA with RF in Table 4. The highest ProbEx, the unbiased measurement for evaluating prediction performance, was obtained with 51 features. On the other hand, the results of feature selection phased conducted by PSO were yielded by XGBoost based on the results given in Table 4. According to the overall results reported in Table 4, the proposed feature selection approach used with BOA provided almost the best results with the smallest number of features.

The proposed approach was applied to a completely unseen external data set explained in the Data Preparation section to meet the fair comparison. As given in Table 1, External data set can be assumed as an imbalanced data set. Since the relative class frequency does not influence ProbEx, most of the analyses and explanations for External data set given in Table 5 were done by ProbEx.

Table 5 Comparison of different classification methods with all features sets and feature subsets obtained by PSO and BOA for External Data

Classification Method	data sets (#of Features)	F1-score	Acc	Precision	Recall	Spec	ProbEx	AUC
DT	Baseline (162)	97.22	95.25	99.33	95.20	74.36	69.56	86.85
	PSO (87)	96.47	93.99	98.16	94.84	72.65	67.49	85.41
	BOA (51)	97.44	95.67	98.50	96.41	81.20	77.60	89.85
ExTrees	Baseline (162)	98.76	97.91	99.83	97.71	88.03	85.75	93.93
	PSO (87)	98.60	97.63	99.83	97.39	86.32	83.72	93.08
	BOA (51)	98.76	97.91	100.00	97.56	87.18	84.74	93.59
LightGBM	Baseline (162)	96.37	93.72	99.67	93.28	63.25	56.53	81.46
	PSO (87)	95.37	91.90	99.83	91.30	51.28	42.58	75.56
	BOA (51)	95.46	92.04	100.00	91.31	51.28	42.59	75.64
RF	Baseline (162)	98.60	97.63	100.00	97.24	85.47	82.71	92.74
	PSO (87)	97.88	96.37	100.00	95.84	77.78	73.62	88.89
	BOA (51)	98.27	97.07	99.83	96.76	82.91	79.67	91.37
XGBoost	Baseline (162)	97.87	96.37	99.83	95.99	78.63	74.62	89.23
	PSO (87)	96.69	94.27	99.83	93.73	65.81	59.54	82.82
	BOA (51)	97.56	95.81	100.00	95.23	74.36	69.59	87.18
Overall	Baseline (162)	97.76	96.18	99.73	95.89	77.95	73.83	88.84
	PSO (87)	97.00	94.83	99.53	94.62	70.77	65.39	85.15
	BOA (51)	97.50	95.70	99.67	95.45	75.39	70.84	87.53

DT: Decision Tree, ExTrees: Extra Trees, RF: Random Forest Acc: Accuracy, Spec: Specificity, ProbEx: Probability Excess, AUC: Area Under Curve

In the model-based analysis, a comparison of the overall metrics in both Table 4 and Table 5 reveals that ExTrees and RF consistently outperform other methods. They achieve high accuracy, precision, and AUC while exhibiting minimal declines in recall and specificity. This suggests that these methods are suitable for handling both the Baseline feature sets and the reduced feature sets without significant performance degradation. In contrast, methods such as LightGBM, although effective with the full set of features, demonstrate greater sensitivity to feature reduction, particularly impacting recall and ProbEx.

The Baseline feature set (162 features) consistently yields slightly better results across all metrics compared to the reduced feature sets (PSO and BOA). However, the differences are generally minimal. This observation indicates that while feature selection may lead to some loss in precision, recall, and other metrics, the trade-off is justified by the reduction in computational complexity and the potential for avoiding overfitting. By eliminating redundant or irrelevant features, the risk of overfitting can be mitigated, which can enhance the generalizability of the model despite minor performance losses in specific metrics.

5. Discussion

Regarding individual results obtained by the methods and the data sets in Table 5, it can be seen that there is no method-data set pair that consistently produces the best results. The results are varied across different methods and data sets, indicating that there is no one-size-fits-all solution. Based on the results presented in Table 5, the highest scores had been obtained by ExTrees. Although there is no observed difference among the results of the ExTrees, outcomes of the data set reduced a third by BOA can be assumed as promising to predict mutagenicity via in-silico methods. To summarize the comparison, the average results of the tree-based classification methods had been calculated. In the light of these averages of the metrics, one can say that when the proposed approach given in Algorithm 1 had been used with BOA, considerably better results had been obtained with reduced data sets both for test and external data sets.

The results achieved by the reduced data set were compared with the results published in 2012 by Xu et al. (Xu et al. 2012) with the almost similar external data set for the sake of completeness of the study. While the external data set used in this study contained 599 mutagens and 117 non-mutagens chemicals, the original data set used in the study of Xu et al. had 614 mutagens and 117 non-mutagens. The accuracies they obtained by these data sets with different fingerprints (descriptors) were from 90.4% to 98%. On the other hand, the range of the accuracy achieved for the external set used in this study by the proposed approach with BOA is from about 91.9% to 97.91%. It is essential to mention that both studies used the same chemicals with mutagenicity information, whereas their number of features and way of calculation is different. Seal et al. (Seal et al. 2012) have generated prediction models using RF classifier for predicting mutagenicity with the data set named Set3, similar to Baseline data set used in this study. The data set used in their study consists of the Bursi and Benchmark data sets explained in the Data Preparation section. According to the outcomes that they published in 2012, they have found the success rate of predicting mutagenicity as 85.15% and precision as 85.2% with 154 descriptors (Seal et al. 2012). Our study shows that results of the data set were reduced into 51 descriptors with BOA and mutagenicity prediction conducted by RF given in Table 4 yielded better results with 86.47% and 88.62% accuracy and precision, respectively.

Each optimization method uses different strategies and metrics. Therefore, selected features can be vary based on the search strategy of the algorithm. Through the hybridization of several optimization techniques with classification algorithms, the suggested method may be able to overcome this variability, eliminate the characteristics of the dataset, and lower the risk of overfitting. To evaluate the effectiveness of the proposed method in terms of overfitting, 5-fold cross-validation was applied alongside the dataset with an external validation test set. The test and validation sets results given in Table 4 and Table 5, respectively, are also a sign that there is no possible overfitting. Moreover, the proposed approach may effectively explore the complex and high-dimensional feature space of the drug toxicity datasets due to the stochastic, adaptive, and global search characteristics of the optimization algorithms.

It is worth mentioning that this approach can be used not only for mutagenicity prediction but also for different problems requiring feature selection and prediction. Since the presented approach can be conducted with any number and kind of metaheuristic optimization algorithms and classification methods, it could be considered a general and flexible framework and a wide range of application fields. It should be noted that, flexibility is the strongest property of proposed hybrid approach. However, the computational complexity is the weakness of the hybrid approach, and it is planned in future studies to overcome this with the new approaches even in high dimensional datasets.

6. Conclusions

Recently, because of the laborious and expensive nature of the drug discovery process, in-silico approaches have played crucial and indispensable roles in the drug approval process. Predicting mutagenicity, which can be defined as the most critical endpoint of toxicity at the early stages of the drug discovery process, is one of the essential steps. This study recommended a flexible approach as an in-silico method both for the early prediction of chemical mutagenicity and reducing the search space into the most effective descriptors. The proposed framework was designed as two sequential phases: feature selection phase through nature inspired optimization algorithms and prediction phase by several statistical and machine learning classification methods. Incorporating metaheuristic algorithms into in-silico studies is not commonly seen in the literature. One of the primary purposes of this study is to conclude whether using the metaheuristic algorithms can be suitable to search the descriptor space in the field of mutagenicity prediction. In order to reach this aim, the butterfly optimization algorithm (BOA) was hybridized with several statistical machine learning algorithms to select the most critical descriptors that are effective in predicting mutagenicity. As mentioned earlier, to the best of our knowledge, no studies are searching for an effective and representative subset of descriptors for mutagenicity estimation through metaheuristic optimization algorithms. To fair comparison, besides BOA, inspired by social butterfly foraging strategy, particle swarm optimization algorithm (PSO), inspired by not a single animal but swarm which is coordinated, were used (Arora and Singh 2019).

Two data sets were used to present the proposed approach: one for selecting the most informative descriptors and modelling; the other for validation. All descriptors were calculated by freely available PaDEL Descriptors software by using SMILES format of the molecules. The original data set contains 162 descriptors. The proposed approach with BOA reduced the number of descriptors to 51, whereas 87 were obtained with PSO. The experimental results present that the outcomes obtained by the BOA have yielded better results, especially with a smaller number of the descriptors sets. In the test data obtained from PaDEL, the highest ProbEx was obtained with the features selected with BOA. While 69.84% ProbEx was obtained with a baseline containing 162 variables in total, due to the model established using 51 variables, approximately 71% ProbEx value was reached with an increase of 1.15% with PaDEL. External Data also achieved the highest ProbEx with Baseline, but the 51 variables selected with BOA performed higher than the 87 variables selected with PSO. As a result, BOA and PSO methods were used for variable selection in the study, and the selected variables were classified using tree-based methods such as DT, ExTrees, RF, LightGBM, and XGBoost. Since no parameter optimization is performed specifically for the data sets, methods that perform highly in PaDEL data may have lower performance in the External data set. Another reason is that while the class distribution is balanced in PaDEL data, the proportion of classes that are non-mutagen in the External data set is lower. Parameter optimization can be performed to increase model performance in future studies.

To ensure completeness of the study, the results were also compared with the results achieved by the studies Seal et al. (2012), Xu et al. (2012), which used similar chemicals. It could be concluded that our approach conducted with nature inspired BOA performed well in terms of accuracy and precisions.

As stated earlier, in-silico studies, i.e., the computational approach to toxicity, has started to gain more attention since predicting mutagenicity at the beginning of the drug design process has been inevitable and is a crucial step to shorten the process and thereby reduce the cost. This study was conducted to present highlight the importance of this approach. The findings in this study suggest that in-silico approaches have a significant role in the drug discovery process by predicting mutagenicity, reducing the search space, and ultimately saving time and resources. The use of metaheuristic optimization algorithms in this context represents a flexible approach that can potentially effective feature selection and prediction in various fields. Further research, including parameter optimization and multi-objective algorithms, can continue to refine and expand upon this methodology. As the part of the future works, a wider range of metaheuristic algorithms and machine learning algorithms can be evaluated to identify the best combination for different drug toxicity endpoints on a larger and more diverse dataset of compounds based on the experimental process, findings and also limitations of the study. It is worth pointing out that, although the computational complexity is a challenge, aiming to address this issue can be also one of the future studies.

In summary, this research demonstrates the potential of combining nature-inspired optimization algorithms with machine learning techniques for feature selection and mutagenicity prediction. The flexible framework presented here can be applied to a wide range of applications requiring feature selection and prediction.

References

Algamal, ZY, Qasim, MK, Lee, MH and Ali, HTM. 2020. High-dimensional QSAR/QSPR classification modelling based on improving pigeon optimization algorithm. Chemom. Intell. Lab. Syst, 206:104170, doi:10.1016/ j.chemolab.2020.104170.

Arora, S, Singh, S. 2019. Butterfly optimization algorithm: a novel approach for global optimization. Soft Comput. 23, 715–34 doi:10.1007/ s00500-018-3102-4.

Bakhtyari, N, Raitano, G, Benfenati, E, Martin, T and Young, D. 2013. Comparison of in silico models for prediction of mutagenicity. Carcinog. Ecotoxicol. Rev, 31(1):45–66, doi:10.1080/10590501.2013.763576

Breiman, L. 2001. Random forests. Mach. Learn, 45:5–32. doi:10.1023/ A:1010933404324.

Çakmak Pehlivanlı, A. and Çakmak, G. 2022. Genotoksik etkiyi belirlemeye yönelik in-silico yaklaşımlar. In Genetik Toksikoloji (Genetic Toxicology), ed. F. Ünal and D. Yüzbaşıoğlu, 475–92. Ankara: Nobel.

Cariello, NF, Wilson, JD, Britt, BH, Wedd, DJ, Burlinson, B and Gombar, V. 2002. Comparison of the computer programs DEREK and TOPKAT to predict bacterial mutagenicity. Mutagenesis 17(4):321-9, doi:10.1093/mutage/17.4.321.

Chen, T and Guestrin, C. 2016. XGBoost. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–94, doi:10.1145/2939672.2939785.

Chu, CSM, Simpson, JD, O'Neill, PM and Berry, NG. 2021. Machine learning predicting Ames mutagenicity of small molecules. Journal of Molecular Graphics and Modelling, 109. doi:10.1016/j.jmgm.2021.108011.

Cover, T. and Hart, P. 1967. Nearest neighbor pattern classification. IEEE Trans. Inf.Theory, 13(1):21–27. doi:10.1109/TIT.1967.1053964.

Fan, D., Yang, H., Li, F, Sun, L, Di, P, Li, W, Tang, Y and Liu, G. 2018. In silico prediction of chemical genotoxicity using machine learning methods and structural alerts. Toxicol, 7(2): 211–20. doi:10. 1039/c7tx00259a.

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

Greene, N, Judson, P, Langowski, J and Marchant, C. 1999. Knowledge-based expert systems for toxicity and metabolism prediction: DEREK, StAR and METEOR. SAR QSAR Environ. Res. 10 2-3, 299-314. doi:10.1080/10629369908039182.

Guan, D, Fan, K, Spence, I and Matthews, S. 2018. QSAR ligand dataset for modelling mutagenicity, genotoxicity, and rodent carcinogenicity. Data Br, 17:876–84, doi: 10.1016/j.dib.2018.01.077.

Gupta, V and Rana, P. 2019. Toxicity prediction of small drug molecules of aryl hydrocarbon receptor using a proposed ensemble model. Turkish J. Electr. Eng. Comput. Sci, 27(4): 2833–49. doi:10.3906/elk-1809-9.

Hansch, C. 1980. Use of quantitative structure-activity relationships (QSAR) in drug design (review). Pharmaceutical Chemistry Journal, 14. doi: 10.1007/BF00765654.

Hansen, K, Mika, S, Schroeter, T, Sutter, A, Laak, AT, Steger-Hartmann, T, Heinrich, N and Müller, KR. 2009. Benchmark data set for in silico prediction of Ames mutagenicity. J. Chem. Inf. Model. 49, 9, 2077–81. doi:10.1021/ci900161g.

Haykin, S. 2011. Neural Networks and Learning Machines. Pearson Education, 3rd ed.

Ho, T. 1998. The random subspace method for constructing decision forests. IEEE Trans. Pattern Anal. Mach. Intell, 20(8):832–44. doi: 10.1109/34.709601.

Honma M, Kitazawa, A, Cayley, A, Williams, RV, Barber, C, Hanser, T, Saiakhov, R, Chakravarti, S, Myatt, GJ, Cross, KP et. al.2019. Improvement of quantitative structure-activity relationship (QSAR) tools for predicting ames mutagenicity: Outcomes of the Ames/QSAR international challenge project. Mutagenesis, 34:41–48. doi:10. 1093/mutage/gey031.

Houssein, E, Honey, M, Oliva, D, Mohamed, W and Hassaballah, M. 2020. A novel hybrid harris hawks optimization and support vector machines for drug design and discovery. Comput. Chem. Eng, 133:106656. doi:10. 1016/j.compchemeng.2019.106656.

Ji, X, Tong, W, Liu, Z and Shi, T. 2019. Five-feature model for developing the classifier for synergistic vs. antagonistic drug combinations built by XGBoost. Front. Genet, 10(JUL):1–13. doi:10.3389/fgene.2019.00600.

John, L, Mahanta, HJ, Soujanya, Y, Narahari Sastry, G. 2023. Assessing machine learning approaches for predicting failures of investigational drug candidates during clinical trials. Computers in Biology and Medicine, Vol.153, 106494. doi: 10.1016/j.compbiomed.2022.106494.

Kazius, J, McGuire, R and Bursi, R. 2005. Derivation and validation of toxicophores for mutagenicity prediction. J. Med. Chem, 48. doi:10.1021/ jm040835a.

Ke, G, Meng, Q, Finley, T, Wang, T, Chen, W, Ma, W, Ye, Q and Liu, T. 2017. LightGBM: A highly efficient gradient boosting decision tree. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Vol. 30, 3149–57. Curran Associates Inc.

Kennedy, J and Eberhart, R. 1995. Particle swarm optimization. pages 1942–48. doi:10.1109/ICNN.1995.488968.

Liao, Q, Yao, J and Yuan, S. 2007. Prediction of mutagenic toxicity by combination of recursive partitioning and support vector machines. Mol. Divers 11, 59–72. doi:10.1007/s11030-007-9057-5.

Mazzatorta, P, Tran, L, Schilter, B and Grigorov, M. 2007. Integration of structure activity relationship and artificial intelligence systems to improve in silico prediction of ames test mutagenicity. J. Chem. Inf. Model. 47, 1, 34–38. doi: 10.1021/ci600411v.

Mirjalili, S and Lewis, A. 2013. S-shaped versus V-shaped transfer functions for binary particle swarm optimization. Swarm Evol. Comput, 9:1–14. 10.1016/j.swevo.2012.09.002.

Mitchell, T. 1997. Machine Learning. McGraw-Hill. New York

Moorthy, N, Kumar, S and Poongavanam, V. 2017. Classification of carcinogenic and mutagenic properties using machine learning method. Comput. Toxicol, 3:33–43. doi: 10.1016/j.comtox.2017.07.002.

Raghavan, N, Amaratunga, D, Nie, AY and McMillian, M. 2005. Class prediction in toxicogenomics, Journal of Biopharmaceutical Statistics, 15:2, 327-41, doi: 10.1081/BIP-200048836

Rifaioglu, AS, Atas, H, Martin, MJ, Cetin-Atalay, R, Atalay,V and Doğan, T. 2019. Recent applications of deep learning and machine intelligence on in silico drug discovery: Methods, tools and databases. Brief. Bioinform, 20(5):1878–1912. doi: 10.1093/bib/bby061.

Seal, A, Passi, A, Jaleel, U, Wild, D and Consortium, O. 2012. In-silico predictive mutagenicity model generation using supervised learning approaches. J. Cheminform. 4(1):10. doi:10.1186/1758-2946-4-10.

Sharma, A, Kumar, R, Varadwaj, P, Ahmad, A and Ashraf, G. 2011. A comparative study of support vector machine, artificial neural network and bayesian classifier for mutagenicity prediction. Interdiscip. Sci. Comput. Life Sci, 3(3):232–239. doi:10.1007/s12539-011-0102-9.

Stevens, S.S. 1986. Psychophysics: Introduction to Its Perceptual, Neural and Social Prospects. 1st ed. Routledge. doi.org/10.4324/9781315127675

Subaş, N and Çakmak Pehlivanlı, A. 2020. İkili parçacık sürü optimizasyonu ve destek vektör makinelerinin hibrit kullanımı ile ilaç keşfi için özellik seçimi. Gümüşhane Üniv. Fen Bilim. Enst. Derg., 11:169–78. doi:10. 17714/gumusfenbil.776329.

Tran, T T V, Surya Wibowo, A, Tayara, H and Chong, KT. 2023. Artificial intelligence in drug toxicity prediction: Recent advances, challenges, and future perspectives. Journal of Chemical Information and Modeling, 63(9):2628–43. doi: 10.1021/acs.jcim.3c00200.

Toropov, AA, Toropova, AP, Raska, I, Leszczynska, D, Leszczynski, J. 2014. Comprehension of drug toxicity: Software and databases. Computers in Biology and Medicine, 45: 20-25. doi: 10.1016/j.compbiomed.2013.11.013.

Tubishat, M, Alswaitti, M, Mirjalili, S, Al-Garage, M, Alrashdan, M and Rana, T. 2020. Dynamic butterfly optimization algorithm for feature selection. IEEE Access, 8:194303–14. doi:10.1109/access.2020.3033757.

Vapnik, V. 1995. The Nature of Statistical Learning Theory. Springer-Verlag, Berlin, Heidelberg.

Webb, SJ, Hanser, T, Howlin, B, Krause, P and Vessey, J. 2014a. Feature combination networks for the interpretation of statistical machine learning models: Application to Ames mutagenicity. J. Cheminform, 6(1):8. doi: 10.1186/1758-2946-6-8.

Webb, SJ, Hanser, T, Howlin, B, Krause, P and Vessey, J. 2014b. Interpretable Ames mutagenicity predictions using statistical learning techniques. In Handbook of abstracts, 6th Joint Sheffield Conference on Chemoinformatics. Qsar2012, 3–4.

White, A, Mueller, R, Gallavan, R, Aaron, A and Wilson, A. 2003. A multiple in silico program approach for the prediction of mutagenicity from chemical structure. Mutat. Res. - Genet. Toxicol. Environ. Mutagen, 539:77–89. doi:10.1016/S1383-5718(03)00135-9.

Wichard, J.D. 2017. In silico prediction of genotoxicity. Food and Chemical Toxicology, 106(Pt B):595-599. doi: 10.1016/j.fct.2016.12.013.

Xu, C, Cheng, F, Chen, L, Du, Z, Li, W, Liu, G, Lee, PW and Tang,Y. 2012. In silico prediction of chemical names mutagenicity. Journal of Chemical Information and Modeling, 52(11):2840–47. doi:10.1021/ci300400a.

Yang, Z, Thomson, R, Mcneil, P and Esnouf, R. 2005. Ronn: the bio-basis function neural network technique applied to the detection of natively disordered regions in proteins. Bioinforma. Orig. Pap, 21(16):3369–3376. doi:10.1093/bioinformatics/bti534.

Yap, C. 2010. Padel-descriptor: an open-source software to calculate molecular descriptors and fingerprints. J. Comput. Chem, 32(4):1466–74. doi:10.1002/jcc.21707.

Zhang, H, Yu, P, Zhang, TG, Kang, YL, Zhao, X, Li, YY, He, JH and Zhang, J. 2015. In silico prediction of drug-induced myelotoxicity by using naïve Bayes method. Mol. Divers, 19(4): 945-53. doi: 10.1007/s11030-015-9613-3.

Zhang, H, Yu, P, Xiang, ML, Li, XB, Kong, WB, Ma, JY, Wang, JL, Zhang, JP and Zhang, J. 2016. Prediction of drug-induced eosinophilia adverse effect by using SVM and naïve Bayesian approaches. Med. Biol. Eng. Comput, 54(2–3):361–9. doi: 10.1007/s11517-015-1321-8.

Zhang, H, Kang, YL, Zhu, YY, Zhao, KX, Liang, JY, Ding, L, Zhang,TG and Zhang, J. 2017. Novel naïve Bayes classification models for predicting the chemical Ames mutagenicity. Toxicol. Vitr, 41:56–63. doi: 10.1016/j.tiv.2017.02.016.

Zhang, J, Mucs, D, Norinder, U and Svensson, F. 2019. LightGBM: An effective and scalable algorithm for prediction of chemical toxicity-application to the Tox21 and mutagenicity data sets. J. Chem. Inf. Model., 59(10):4150–58. doi: 10.1021/acs.jcim.9b00633.

Zheng, M, Liu, Z, Xue, C, Zhu, W, Chen, K, Luo, X and Jiang, H. 2006. Mutagenic probability estimation of chemical compounds by a novel molecular electrophilicity vector and support vector machine. Bioinformatics. 22(17):2099-106. doi: 10.1093/bioinformatics/btl352.

Research Article
Journal of Intelligent Systems: Theory and Applications 7(2) (2024) 129-144
DOI: 10.38016/jista.1516453

Classification of Quality Defects using Multivariate Control Chart with Ensemble Machine Learning Model

Deniz Demircioğlu Diren[1*], Semra Boran[2]

[1] Distance Education Research and Application Center, Sakarya University, Sakarya, Türkiye

[2] Faculty of Engineering, Industrial Enginering Department, Sakarya University, Sakarya, Türkiye

ddemircioglu@sakarya.edu.tr, boran@sakarya.edu.tr

Abstract
Multivariate control charts enable to monitor processes affected by more than one variable. But, when the process is out of control, it cannot detect which variable is causing it. It is an important requirement to know which variables in the process need corrective actions. In this study, a machine learning-based model is proposed to predict the variable/s that make the process out of control. For this purpose, ensemble algorithms, which are known to have higher prediction performance than single algorithms, were preferred. Because it is aimed to determine the variable(s) that cause the process to be out of control in the most accurate way. It is thought that a classification model in which ensemble algorithms are used together can increase the prediction accuracy. The model, which has not been encountered before in a quality control problem, was applied to a real problem and 98.06% classification accuracy was achieved. Another benefit is that it can predict the variable/variables that make the process uncontrolled without the need for multivariate control charts.

Keywords: Multivariate control chart, Machine learning, Ensemble of ensemble algorithm, Hotelling T^2 chart, Mason-Young-Tracy method.

Çok Değişkenli Proses Kontrol Grafiği ve Topluluk Makine Öğrenme Modeli Kullanılarak Kalite Kusurlarının Sınıflandırılması

Öz
Çok değişkenli kontrol diyagramları birden fazla değişkenin etki ettiği süreçlerin izlenmesine olanak sağlamaktadır. Ancak süreç kontrol dışında olduğunda hangi değişkenin buna neden olduğunu tespit edilememektedir. Süreçteki hangi değişkenlerin düzeltici faaliyetlere ihtiyaç duyduğunu bilmek önemli bir gerekliliktir. Bu çalışmada süreci kontrolden çıkaran değişken/değişkenleri yüksek doğrulukla belirlenmesi tahmin etmek için makine öğrenmesi tabanlı bir model önerilmiştir. Bu amaçla tekli algoritmalara göre daha yüksek tahmin performansına sahip olduğu bilinen topluluk algoritmaları tercih edilmiştir. It is thought that a classification model in which ensemble algorithms are used together can increase the prediction accuracy. Daha önce bir kalite kontrol probleminde rastlanmayan model, gerçek bir probleme uygulanmış ve %98,06 sınıflandırma doğruluğu elde edilmiştir. Ayrıca bir diğer faydası da çok değişkenli kontrol grafiklerine ihtiyaç duymadan süreci kontrolden çıkaran değişken/değişkenleri tahmin edebilmesidir.

Anahtar Kelimeler: Çok değişkenli kontrol grafiği, Makine öğrenmesi, Topluluk algoritması topluluğu, Hotelling T^2 grafiği, Mason-Young-Tracy yöntemi.

1. Introduction

In order to produce quality products and ensure their sustainability, processes must be constantly monitored. The causes of out-of-control situations encountered while monitoring the processes should be determined as accurately and quickly as possible and corrective actions should be implemented. Since the products used today have a much more complex structure, the production processes should be evaluated according to their many features (Robert, 2002). While traditional control charts deal with a single measurable product feature (variable), multivariate control charts have the feature of being tools that can handle multiple variables simultaneously (Montgomery, 2009). Thus, time and cost savings are

[*] Corresponding Author.
E-mail: ddemircioglu@sakarya.edu.tr

Recieved : 15 Jul 2024
Revision : 7 Aug 2024
Accepted : 26 Aug 2024

achieved. In addition, these control charts also enable the evaluation of the relationship between the variables (Hotelling, 1947; Woodall, 1985; Lowry, 1992). In addition to this advantage, the most criticized feature of the charts is that it cannot detect which variable/s caused it in case of out-of-control signals (Aparasi, 2006). However, this causes very important problems because it is necessary to know which variable(s) corrective action should be applied in order to control the process. Because wrong estimation of variable lead to loss of time, increase in finances and worst of all, poor quality products. For this, traditional methods are not sufficient and new methods are needed. The leading of these is the Mason, Young, Tracy Decomposition (MYT) method, which has been specially developed for quality control charts. Principal component analysis and discriminant analysis are also used for similar purposes. (Jackson,1985; Rao et al., 2013; Pei et al., 2006; Hawkins, 1991; Mason et al., 1997; Das and Prakash, 2008; Li et al., 2008; Agog et al., 2014; Joshi and Patil, 2022). In addition to the mentioned statistical methods, it is seen that machine learning algorithms are frequently used in recent years (Aparasi, 2006; Niaki and Abbasi, 2005; Chen and Wang, 2004; Cheng and Cheng, 2008; Song et al., 2017; Shao and Lin, 2019; Du et al., 2012; Asadi and Farjami, 2019; Ahsan et al., 2020; Sabahno & Amiri, 2023). However, statistical methods have weaknesses such as not being able to make predictions for new data and not measuring the accuracy of the results with various criteria. Machine learning algorithms are more preferred because they have features to eliminate these weaknesses. In this study, a new ensemble machine learning model developed using the results obtained from Hotelling T^2 and MYT methods is presented in order to detect the variable(s) causing out-of-control situations. With this model, it is aimed to determine the variable(s) that cause out-of-control situations as accurately as possible. It is known that ensemble machine learning algorithms provide more accurate predictions than single algorithms (Jiang and Song, 2017; Asadi and Farjami, 2019). For this reason, the bagging and boosting ensemble algorithms in the classification model developed in the study were combined with the stacked generalization algorithm, which is another ensemble algorithm, and the ensemble structure ensemble was used. Thus, the variable(s) causing the out-of-control situation were determined in the most accurate way. Model data were obtained with Hotelling T^2, which is a multivariate control chart, and MYT method, which was specially developed for the chart. In order to determine the algorithm to be used in the model, Decision Trees (DT), Naive Bayes (NB), K-Nearest Neighbor (KNN), Multi Support Vector Machines (M-SVM) and Artificial Neural Networks (ANNs) were used, which are among the most basic single algorithms. Since the aim was to increase prediction accuracy, the algorithm that was most successful in single uses was chosen first. Then, ensemble models were developed with this algorithm using bagging and boosting.

The subsequent of the article is organized as follows. In Section 2, a literature review will be conducted. After explaining the methods in Section 3, the proposed model will be presented in Section 4. Then, the implementation will be carried out in Section 5 to carry out the experimental study of the model. The article concludes with Section 6, where discussion and conclusion is presented.

2. Literature Review

There are many studies in the literature about the determination of the variable(s) that cause the out-of-control situation, using statistical and machine learning methods.

The most frequently used method in the literature for multivariate control charts is the Mason Young Tracy (MYT) decomposition method (Robert, 2002). This method, which was developed by Mason et al., (1995), was designed specifically for the Hotelling T^2 control chart, based on principal component analysis (Mason et al., 1995; Özel, 2005). There are studies in many different areas where MYT is used (Çetin and Birgören, 2007; Parra and Loaiza, 2003; Ulen and Demir, 2013; Boullosa et al., 2017; Yilmaz, 2012).

Studies in which machine learning techniques are used to determine the variables that cause the out-of-control situation are examined in two classes as studies in which basic algorithms are used individually and as an ensemble.

Studies using single algorithms to detect variables that cause out-of-control situations have been encountered for many years. In two separate studies by Chen and Wang (2004) and Niaki and Abbasi (2005), an artificial neural network-based model was developed for the X^2 chart and presented by evaluating its successful performance. In the study performed by Aparisi et al. (2006), accuracy analysis of MYT method and neural network was performed in terms of classification. According to the results; It has been seen that the accuracy performance of the designed neural network is better than the accuracy performance of the MYT method (Aparisi et al., 2006). In the application by Cheng and Cheng (2008), which aims to detect variables with Artificial Neural Network (ANN) and Support Vector Machine (SVM), the performance of SVM was found to be similar to ANN. In addition, it has been stated that the ANN algorithm has weaknesses such as the large number of control parameters and the difficulty of applying steps. In another study, Li et al. (2013) compared the optimized SVM approach with the developed ANN for the estimation of the shift magnitude in the process. As a result, the best performance of the SVM approach has been demonstrated. Huda et al. (2014) developed an ANN-based model that does not need expert knowledge and requires little numerical computation. The results showed that the proposed approach is successful and easy to implement. Song et al. (2017) proposed a sample-based Navie Bayes (NB) method to interpret out-of-control situations. As a result of the performance comparisons, it was stated that the developed method outperformed other statistical techniques. In the study by Shao and Lin (2019), ANN-based classification model was developed in a multivariate process with variance shift. The performance of this model is compared with ANN, SVM and multivariate adaptive regression classifier. As a result, it was stated that the developed model was more successful. Bersimis et al. (2022) an ANN-based model was developed that uses the results of some analytical methods as input for the detection of uncontrolled variables. According to the results obtained, very successful results were obtained with the developed model. In another study conducted by Rakhmawan et al (2023), the Hotelling T^2 control chart was optimized with the decision tree model. It has been stated that this is a solution that can be used to obtain accurate predictions.

There are studies where ensemble algorithms are used to detect variables that cause out-of-control situations. In the study by Guh and Shiue (2008), a simple and effective model obtained by sequentially combining the Decision Tree (DT) classification algorithm is proposed to detect the mean shifts in multivariate control charts. Experimental results show that the learning speed of the proposed model is much faster than an ANN-based model. For the same purpose, an ANN-based ensemble model was developed by Yu et al. (2009). The results of the study, which produced data according to 5 different shift sizes from the mean for each variable by simulation, are presented that the proposed model outperforms the use of single ANN in terms of average running length (ARL). In the study by Alfaro et al. (2009), ensemble trees have proven to be a very powerful tool for classification accuracy. Du et al. (2012) classified the causes of mean shifts in the multivariate process with the multiclass bagging ensemble SVM algorithm. The performance of the model evaluated according to the accuracy criterion with a real application has been proven to be effective. Similarly, the approach developed in the study by Cheng and Lee (2012) using the bagging ensemble SVM algorithm is compared with the traditional decomposition method and its performance is seen to be more successful. Yang (2015) concluded that the proposed artificial neural network ensemble model is a more effective approach in diagnosing out-of-control situations than other approaches in the literature. In the study by Jiang and Song (2017), which developed an ensemble model by combining decision trees in parallel, it was proven that the classification performance of the ensemble learning method was better. Another study in which decision trees were applied as an ensemble was carried out by Asadi and Farjami (2019). In the study, a structure with four classifiers in which decision trees are connected sequentially and a Monte Carlo simulation are used. The developed model ARL functions were compared according to accuracy, precision and precision criteria. The results showed better performance of the community DT construct. In the research conducted by Alfaro et al. (2020), the random forest method was used to detect out-of-control situations. This method has been compared with ANN and it has been stated that the random forest method is more successful when there is small and medium correlation between variables.

In this study, a ensemble algorithm is proposed in which bagging and boosting ensemble algorithms are combined. Based on the stacked generalization algorithm, this model was used to combine the power of other ensemble algorithms to detect variables that cause out-of-control in a multivariate process. The decision of the basic single algorithm to be used in the bagging and boosting ensemble algorithms was also made according to the high accuracy rate.

3. Methods

3.1. Hotelling T^2 Control Chart

Hotelling T^2 control chart was developed by Hotelling in 1947 to monitor the related p number of variables simultaneously (Montgomery, 2009). The chart is formed by scheduling the T^2 statistic, which is a statistical distance measure based on a multivariate normal distribution (Çetin and Birgören, 2007). In case the sample size is 1, the steps of the control chart are as follows. For each sample, the T^2 statistic is calculated with the help of Equation (1) according to p number of variables.

$$T^2 = (X - \bar{X})'S^{-1}(X - \overline{X}) \tag{1}$$

Where, X is variable, $\bar{X}$ is sample mean vector and S is the sample covariance. While the upper control limit (UCL) for the first phase of the multivariate control chart is calculated according to Equation (2), the lower control limit (LCL) is taken as the zero line as seen in Equation (3).

$$UCL = \frac{(m-1)^2}{m} \beta_{\alpha,p/2,(m-p-1)/2} \tag{2}$$

$$LCL = 0 \tag{3}$$

Where, m expresses the upper α percentage point of the beta distribution with the parameters $\beta_{\alpha,p/2,(m-p-1)/2}$ including the number of samples (Montgomery, 2009).

In order to use the Hotelling T^2 control chart, some assumptions must be met. These assumptions are conformity to multivariate normal distribution, linearity, absence of autocorrelation, variance covariance equality (homogeneity). If there are (s) not provided by the assumptions, the necessary conversion actions should be applied.

3.2. Mason Young Tracy (MYT) Decomposition Method

This method was developed by Mason, Young and Tracy in the 1990s to detect out-of-control variables by splitting the Hotelling T^2 statistic into two orthogonal parts, conditionally and unconditionally. In this method, firstly, the operated and operated are defined continuously and calculations are made. Then possible MYT decompositions are shown, and finally, similar values are calculated in periods and comments are made about the out-of-control variables (Mason et al., 1995).

T^2 statistic in Equation 1 is formed by combining conditional and unconditional terms as seen in Equation 4.

$$T^2 = T^2_{p-1} + T^2_{p.1,\ldots,p-1} \tag{4}$$

Here, the part shown in Equation (5) expresses the unconditional terms.

$$T^2_{p-1} = \left(X_i^{(p-1)} - \bar{X}^{(p-1)}\right)' S_{XX}^{-1} \left(X_i^{(p-1)} - \bar{X}^{(p-1)}\right) \tag{5}$$

Where, $\bar{X}^{(p-1)}$ is the mean vector of n multivariate observation values of the first (p-1) variable. S_{XX} is the (p-1)*(p-1) basic submatrix of S.

The part shown in Equation (6-9) expresses the conditional terms.

$$T^2_{p.1,\ldots,p-1} = \frac{x_{ip} - \bar{x}_{p.1,\ldots,p-1}}{s^2_{p.1,\ldots,p-1}} \tag{6}$$

$$\bar{X}_{p.1,\ldots,p-1} = \bar{X}_p + b'_p \left(X_i^{(p-1)} - \bar{X}^{(p-1)}\right) \tag{7}$$

Where $\bar{X}_p$ is the sample mean of n observation values of the pth variable.

$b_p = S_{XX}^{-1} s_{xX}$ is the dimensional vector that estimates the regression coefficients of the p-th variable in the first p-1 variable.

$$s^2_{p.1,\ldots,p-1} = s_x^2 - s'_{xX} S_{XX}^{-1} s_{xX} \tag{8}$$

$$S = \begin{bmatrix} S_{XX} & s_{xX} \\ s'_{xX} & s_x^2 \end{bmatrix} \tag{9}$$

Where, s_{xX} is the vector of covariance between variables, s_x^2 is the variance of the variable p.

3.3. Machine Learning Algorithms

Machine learning is a technology developed to enable machines to be intelligent, enabling systems to learn directly from examples, data and experiences (The royal society, 2017). These technologies enable machines to make predictions, perform clustering, extract association rules or make decisions from a given data set (Mohammed et al., 2016). It is possible to examine algorithms in two classes, as single and ensemble, according to their usage structure.

3.3.1. Single machine learning algorithms

In single algorithms, only one algorithm is run and the results are obtained accordingly. In the study, DT, NB, ANN, SVM, KNN algorithms will be discussed.

A Decision Tree (DT): DT has a tree structure consisting of nodes. These nodes are called root, intermediate and leaf nodes according to their purpose (Maimon and Rokach, 2010). The working steps of the algorithm first start from the root. Then it continues by branching from the intermediate node to the leaf node. Classes in the tree are represented by leaves, and there is only one path to each leaf (Bilgin,2018; Maimon and Rokach, 2010; Han et al., 2012; Mitchell, 2014; Agrawal and Imielinsk, 1993; Utgoff et al., 1997). The samples are classified from the root of the tree to a leaf according to the result of the tests carried out along the way. These results can then be combined into a rule by taking the class estimate of the leaf as the class value (Maimon and Rokach, 2010). This structure, which can be re-represented with IF-THEN rule sets for easy understanding by the user (Mitchell,2014), can contain both nominal and numerical properties. Commonly used criteria for determining the root node feature are Information Gain, Gini index, Gain Ratio (Maimon and Rokach, 2010).

Naive Bayes (NB): NB is used when there is leading knowledge and provides a probabilistic approach to logical inference. It aims to combine the value from the sample with the leading information. This algorithm ignores the relationships between the inputs and reduces a multivariate distribution to multiple univariate distributions, as seen in Equation (10) (Alpaydın, 2012).

$$p(x|C) = \prod_{j=1}^{d} p\left(x_j \middle| C\right) \tag{10}$$

Here; P(X): Probability of X (independent), P(Y): Probability of Y (independent), P(X | Y): Probability of X occurring when Y has occurred and P(Y | X): Probability of Y occurring when X has occurred.

K-Nearest Neighbor (KNN): KNN is based on classification with the nearest neighbors approach (Han et al., 2012). The number of neighbors (k) is determined by the user. In order to find the location of the nearest neighbors of a sample, a distance function or criteria such as Euclidean, Manhattan and Minkowski Distance, which measure the similarity between two samples, are used (Bilgin, 2018). Euclidean Distance shown in Equation (11) was used in the study. Where p and q are two examples compared.

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^{k} (p_i - q_i)^2} \tag{11}$$

Here p and q are two examples compared.

Artificial Neural Network (ANN): ANN is an important classification method that includes parallel computation programs that work similar to the human brain. Multilayer Perceptron (MLP), which is the most commonly used artificial neural network model, consists of three layers: input, hidden and output. While the number of process elements in the input and output layer is determined according to the problem, the number of elements in the hidden layer is determined by trial and error in order to achieve the best performance. The weights showing the importance of the information are determined randomly at the beginning (Öztemel, 2003). Inputs are converted to output with the activation function (Yadav et al., 2015).

Multi-Class Vector Machine (M-SVM): The SVM method developed by Cores and Vapnik (1995) is used for two-group classification and prediction problems of both linear and non-linear data. Its working principle is based on transforming the size of the data, determining decision surfaces and dividing it into two classes in the most appropriate way. When the number of classes is more than two, multi-class vector machines should be used. There are three options for this algorithm.

Here w is the weight vector, x is the sample and r is the class of the data. The size of this interval is very important for the accuracy of classification. When $r^t = +1$ and $r^t = -1$.

1. When K>2, K two class problems are defined and K different separators distinguish each class from other classes; i=1,….,K support vector machine is trained. Here, while training the parser, the samples from the class C_i are classified as +1, and the samples from the class C_k k≠1 are classified as -1. All values are calculated and the largest one is selected.
2. The problem is divided into multiple linear subproblems. The algorithm for this is to train with K(K-1)/2 discriminant binary classifiers, similar to two-class SVM.
3. In this option, a single multi-class optimization problem that includes all classes is considered as seen in Equation (12).

$$\min \frac{1}{2} \sum_{i=1}^{K} \|w_i\|^2 + C \sum_i \sum_t \xi_i^t \tag{12}$$

where the constraints are as seen in Equation (13, 14).

$$w_{z^t} x^t + w_{z^t 0} \geq w_i x^t + w_{i0} + 2 - \xi_i^t, \forall i \neq z^t \tag{13}$$

$$\text{and } \xi_i^t \geq 0 \tag{14}$$

Although this option is a very good approach, it is less preferred than other options in terms of usage due to processing load and time.

3.3.2. Ensemble machine learning algorithms

Ensemble algorithms are predictive models created by combining multiple algorithms of the same or different types with various methods in different ways (Rokach, 2010). It is aimed to achieve higher prediction accuracy with ensemble algorithms than single algorithms.

Ensemble algorithms can be created as dependent /independent and homogeneous/heterogeneous. In the dependent method, the output of one classifier is used by the next classifier. Thus, it is possible to take advantage of the knowledge produced in previous iterations to guide learning in the next iterations. In independent methods, each classifier is created independently and its outputs are combined (Maimon and Rokach, 2010). In dependent methods, algorithms are connected in series with each other, while in independent methods, algorithms are connected in parallel. The basis of parallel ensemble methods is to use independence between single algorithms, since classification and prediction error can be significantly reduced by combining independent base learners (Zhou, 2012).

In addition to combining the algorithms dependently and independently, there are ensemble algorithms that are obtained homogeneously by using the same single algorithm and heterogeneously by using different single algorithms. The classification of ensemble algorithms according to the merging principles is given in the Figure 1 (Zhou, 2012; Gowda et al., 2018).

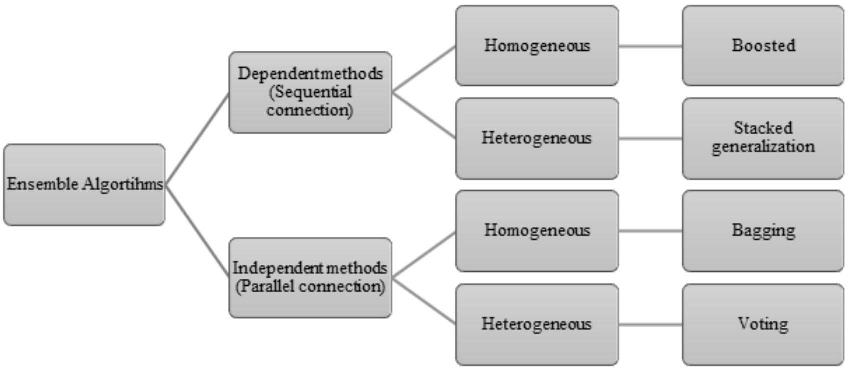


Figure 1. Types of ensemble algorithms

The bagging algorithm method was developed by Breiman (1996) and is the oldest and simplest ensemble algorithm. This method, which is based on combining basic learners in parallel, is a method that can be used with multiple classes (Zhou, 2012; Gowda et al, 2018; Zhang and Ma, 2012).

The boosting ensemble method is based on the principle that algorithms use the output of the previous algorithm as input, and the algorithms are connected in series. In this method, each classifier is affected by the performance of the previous algorithm and gives more importance to classification errors made by previously created classifiers (Rokach, 2010). When the number of classes is more than two, the AdaBoost method, which is the most preferred boosting method, is used (Zhou, 2012).

The stacked generalization method is a meta-learning based ensemble algorithm. Based on the predictions and correct answers of the basic learning algorithms, a meta-learner is trained (Onan, 2018). Here, the basic idea is to train the first-level learners using the original training dataset and then create a new dataset to train the second-level learner in which the outputs of the first-level learners are considered as the input features. First-level learners are often produced by applying different learning algorithms, and therefore stacked method are often heterogeneous (Zhou, 2012). The second-level metadata set consists of the predictions of all algorithms (Onan, 2018).

3.3.3. Performance criteria of machine learning models

In the study, the variable(s) that cause the out-of-control situation are determined by classification. For this reason, performance criteria such as accuracy, classification error, sensitivity and kappa statistics used in the classification problems of the learning performances of the developed models were evaluated. Performance criteria are as in Table 1 (Hossin and Sulaiman, 2015).

Where, g_{pi} is the true number of positives in class i, g_{ni} is the actual number of negatives in class i, y_{pi} is the number of false positives in class i, y_{ni} is the number of false negatives in class i, h_M is the macro mean of sensitivity, k_M is represents the macro average of precision.

Another criterion, the Kappa statistic, evaluates the classification accuracy by taking into account the chance factor in the probability of a correct guess. It is calculated as seen in Equation (15) (Lantz, 2013).

$$k = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \tag{15}$$

Where, $\Pr(a)$ and $\Pr(e)$ represent the agreement ratio between the actual and expected values, the classifier and the actual values, respectively. Kappa values are commonly interpreted as; Bad estimate = less than 0.20, Acceptable estimate = 0.20 – 0.40, Intermediate estimate = 0.40 to 0.6, Good estimate = 0.60 to 0.80, Very good estimate = 0.80 – 1.00.

3.3.4. Handling Imbalanced Dataset

If the classes in the dataset are not approximately equally represented, the dataset can be eliminated the imbalanced. The performance of machine learning algorithms is often based on predictive accuracy. However, when data are unbalanced, often the majority class is predicted with little error, while the minority class(es) cannot be predicted. In this case, it can be said that using predictive accuracy would be misleading. Class imbalance in the data is addressed in two ways. The first is that it assigns different weights to the training examples. The other is to resample the original dataset by either oversampling the minority class and/or undersampling the majority class (Chawla et al., 2002). Synthetic Minority Oversampling (SMOTE) method, widely used for resampling, is a sampling technique that produces synthetic samples from the minority class. This method, which synthetically equates the number of data in the minority class to the number of data in the majority class, is used to obtain a training set with a balanced or nearly balanced class.

Table 1. Performance criteria based on confusion matrix for classification in multi-class problems

Performance Criteria	Formula	Description
Average Accuracy	$\frac{\sum_{i=1}^{l} \frac{g_{pi} + g_{ni}}{g_{pi} + y_{ni} + y_{pi} + g_{ni}}}{l}$	Average effectiveness of classes
Average Error Rate	$\frac{\sum_{i=1}^{l} \frac{y_{pi} + y_{ni}}{g_{pi} + y_{ni} + y_{pi} + g_{ni}}}{l}$	Average error rate of classes
Average Sensitivity	$\frac{\sum_{i=1}^{l} \frac{g_{pi}}{g_{pi} + y_{pi}}}{l}$	Average of precision per class
Average Precision	$\frac{\sum_{i=1}^{l} \frac{g_{pi}}{g_{pi} + y_{ni}}}{l}$	Precision average per class
Average F measurement	$\frac{2 * h_M * k_M}{h_M + k_M}$	F measurement per class

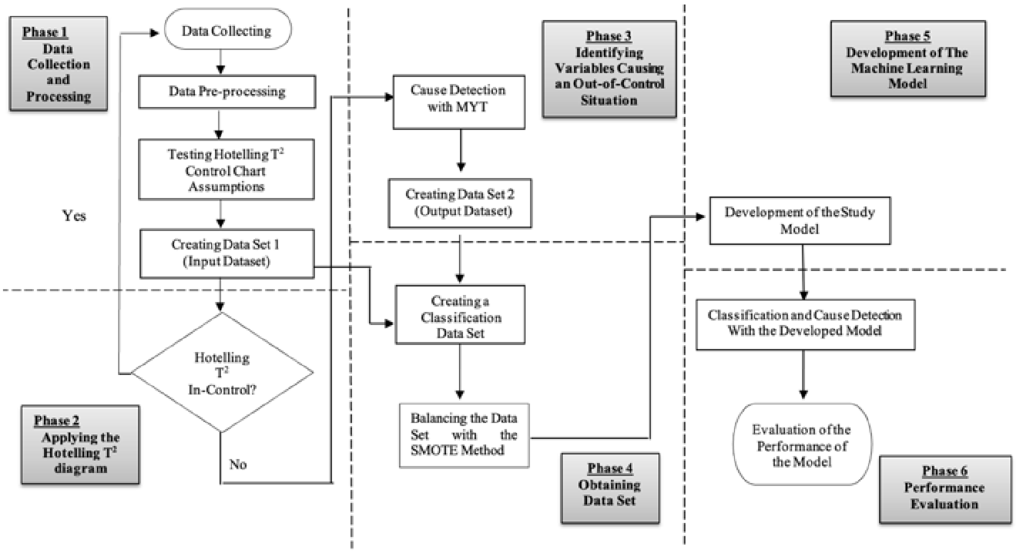

Phase 1
Data Collection and Processing
Data Collecting
Data Pre-processing
Testing Hotelling T² Control Chart Assumptions
Yes
Creating Data Set 1 (Input Dataset)
Hotelling T² In-Control?
No
Phase 2
Applying the Hotelling T² diagram
Cause Detection with MYT
Creating Data Set 2 (Output Dataset)
Creating a Classification Data Set
Balancing the Data Set with the SMOTE Method
Phase 3
Identifying Variables Causing an Out-of-Control Situation
Phase 4
Obtaining Data Set
Development of the Study Model
Classification and Cause Detection With the Developed Model
Evaluation of the Performance of the Model
Phase 5
Development of The Machine Learning Model
Phase 6
Performance Evaluation

Figure 2. Architecture of the proposed model

SMOTE samples are linear combinations of two similar samples from the minority class and are obtained by Equation (16) (Blagus and Lusa, 2013).

$$s = x + u\,(x^R - x) \tag{16}$$

where x^R and x are two similar classes, x^R, is randomly selected from among the five closest minority classes of x. u is a random number between 0 and 1.

4. Proposed Model

The aim of the study is to develop an ensemble model to identify the causes of out-of-control situations in quality processes with the highest accuracy, is shown in Figure 2. The architecture consists of six phases. The phases can be stated as Data Collection and Processing, Applying the Hotelling T^2 chart, Identifying variables causing an out-of-control situation, Obtaining Data Set, Development of The Machine Learning Model and Performance evaluation.

The steps involved in the phases and the proposed model are described in detail below in Figure 2.

Phase 1. Data Collection and Processing: At this phase, data is collected about the examined properties of the manufacturing part. Before analyzing the data set, it should be checked whether it contains outlier, incomplete or inconsistent data, and if there are such cases, the data preprocessing process should be performed (Şişci et al., 2022).

Phase 2. Applying the Hotelling T^2 Chart: At this phase, it will be checked whether the T^2 statistic is suitable for linearity, normal distribution, autocorrelation and variance-covariance equality assumptions so that the data set can be used in the Hotelling T^2 control chart. Since the Hotelling T^2 control chart with a sample size of one is used in the study, the variance-covariance assumption is invalid and there is no need to check this assumption. With the linearity assumption, it is investigated whether there is a desired linear relationship between the two variables. For this, the Pearson correlation coefficients between the two variables should be calculated and compared with the level of significance. If this coefficient is greater than the significance level, there is a linearity relationship. According to the assumption of conformity to the multivariate normal distribution, each of the variables must be suitable for the normal distribution. With the Kolmogorov-Smirnov test, the conformity of the measurement values to the normal distribution is tested. After the suitability of all the variables to the normal distribution has been proven, the suitability of all the variables to the normal distribution should be evaluated with the Henze-Zirkler's test. It should be tested with the Box-Ljung statistic to determine whether there is autocorrelation between the autocorrelation assumption and the variables. After checking all assumptions, a Hotelling T^2 chart is created according to Equation (1).

Phase 3. Identifying Variables Causing an Out-Of-Control Situation: At this phase, the variable(s) that cause the samples outside the upper control limits determined by Hotelling T^2 to be out of control will be determined by MYT decomposition method.

Phase 4. Obtaining Data Set: At this phase, the inputs and outputs are brought together to obtain the data set. Variable measurement values constitute the input, and the variable classes belonging to the out-of-control situations obtained in the MYT results constitute the output. Inputs are obtained in the first phase, and outputs are obtained in the second and third phases. SMOTE was used to eliminate the imbalance caused by the difference in the data numbers of the classes in the data set. In the data set used in the proposed model, similar to other studies in the literature (Alfaro et al., 2009; Jiang and Song, 2019), only out-of-control situations are considered.

Phase 5. Development of the Machine Learning Model: After the data set to be used in the model is obtained, the algorithm to be used in the developed model will be selected. The model consists of the following steps:

1. By applying single machine learning algorithms, the most successful algorithm is selected according to the performance criteria.
2. Combining this selected single machine algorithm with bagging and boosting algorithms in parallel and sequentially.
3. Developing the two ensemble algorithms obtained in the second step by combining them with another ensemble algorithm, the stacked generalization method. The hybrid ensemble model, which is based on the combination of ensemble algorithms in order to increase the prediction performance, will be designed as seen in Figure 3.

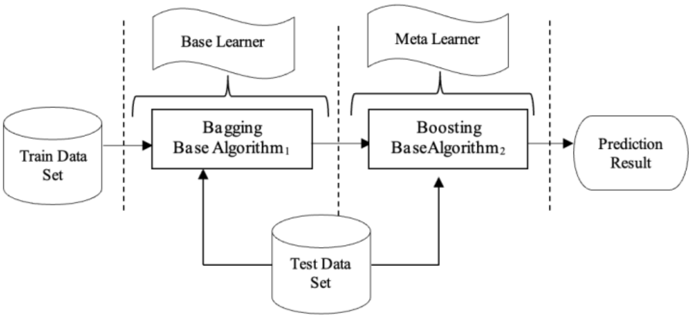


Figure 3. Architecture of the proposed model

Phase 6. Performance Evaluation: The proposed model will be trained with the dataset and its performance will be evaluated according to various criteria. If the evaluation results are found successful, the suitability of the model will be decided. For performance comparisons of classification algorithms, criteria such as accuracy, sensitivity, precision and kappa statistics were used.

5. Implementation of Proposed Model

In order to prove the validity of the proposed model, a real-life problem has been applied in the steel hydraulic pump cover production process of an automotive supplier operating in Turkey. The 3D view of the hydraulic pump cover part is shown in Figure 4. In addition, as seen Table 2, 8 variables that determine the quality of the part were determined by quality experts.

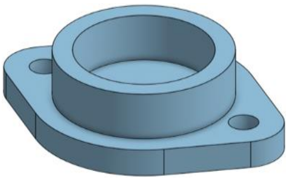


Figure 4. Hydraulic pump cover

Definition of variables, mean and specification values are as in Figure 4.

5.1. Data Collection and Processing

Data is obtained through measurements made during manufacturing. Measurements are made by taking a single sample per hour from the manufacturing process. 26700 measurement values were taken as basis in the study. Each measurement value constitutes a sample in the data set. Outlier, extreme or missing values in the data set were first examined in terms of variable and sample suitability through data pre-processing steps and it was determined that there were no data with undesirable characteristics. However, it was determined that the measurement results for some variables were missing in three samples. Therefore, these three samples were eliminated and quality evaluation was carried out on the remaining 26697 samples.

5.2. Hotelling T^2 Control Chart Implementation

Before applying the Hotelling T^2 control chart, it was checked whether the data met the assumptions regarding the T^2 statistics.

Table 2. Definition of variables

Variables	Definition	Specification Value (mm)	Tolerance (mm)
x_1	1. Hole Diameter	30	±0,2
x_2	2. Hole Diameter	30	±0,2
x_3	Large Outside Diameter	210	±0,5
x_4	Distance Between Holes	230	±0,2
x_5	Cheek Height	21	±0,5
x_6	Cheek Outer Diameter	180	±0,1
x_7	Cheek Inner Diameter	140	±0,5
x_8	Cover Wall Thickness	27	±0,5

Table 3. Correlation matrix between variables (initial case)

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1: 1.hole diameter	P.C	1	-0,75**	0,01	0,63**	0,048**	-0,004	0,062**	0,007
	Sig.		0,000	0,881	0,000	0,000	0,520	0,000	0,264
x_2: 2.hole diameter	P.C		1	-0,002	0,18**	-0,008	-0,001	-0,048**	0,004
	Sig.			0,725	0,003	0,195	0,852	0,000	0,562
x_3: large outer diameter	P.C			1	-0,010	-0,008	0,001	-0,001	0,008
	Sig.				0,104	0,184	0,874	0,881	0,183
x_4: distance between holes	P.C				1	-0,019**	-0,009	-0,038**	0,006
	Sig.					0,002	0,162	0,000	0,344
x_5: cheek height	P.C					1	-0,21**	0,037**	-0,004
	Sig.						0,001	0,000	0,508
x_6: cheek outer diameter	P.C						1	-0,011	0,001
	Sig.							0,065	0,907
x_7: cheek inner diameter	P.C							1	-0,003
	Sig.								0,606
x_8: cover wall thickness	P.C								1
	Sig.								

** Correlation significant at 0.01 level

- **Linearity:** Pearson coefficient was calculated for binary variables to test the linearity assumption. The evaluation result is summarized in Table 3. As can be seen from the table, it is understood that two of the variables (large outer diameter (x3) and cover wall thickness (x8)) have no relationship with any other variable.

For this reason, there was no need to evaluate it with a multivariate control chart. Since these variables are unrelated, they can be handled separately with univariate control charts. Large outer diameter (x_3) and cap wall thickness (x_8) variables were removed from the data set and the linearity assumption was repeated for six variables. The pearson correlation coefficient (P.C) values calculated to evaluate the relationships between six variables are shown in Table 4. When the significance levels of the remaining six variables are examined, they are generally seen to be significant, that is, there is a linear relationship.

- **Assumption of suitability for multivariate normal distribution:** The normal distribution suitability test results obtained for 6 quality variables are given in Table 5. It can be said that the p value for all variables is greater than 0.05 and therefore all variables individually comply with normal distribution.

Table 4. Correlation matrix between variables (final situation)

		x_1	x_2	x_3	x_4	x_5	x_6
x_1: 1.hole diameter	P.C	1	-0,75**	0,63**	0,048**	-0,004	0,062**
	Sig.		0,000	0,000	0,000	0,520	0,000
x_2: 2.hole diameter	P.C		1	0,18**	-0,008	-0,001	-0,048**
	Sig.			0,003	0,195	0,852	0,000
x_3: distance between holes	P.C			1	-0,019**	-0,009	-0,038**
	Sig.				0,002	0,162	0,000
x_4: cheek height	P.C				1	-0,21**	0,037**
	Sig.					0,001	0,000
x_5: cheek outer diameter	P.C					1	-0,011
	Sig.						0,065
x_6: cheek inner diameter	P.C						1
	Sig.						

** Correlation significant at 0.01 level

Table 5. Univariate normal distribution results

Test	Variable	KS value	P value
Kolmogorov-Smirnov	x_1	0,004	>0,150
Kolmogorov-Smirnov	x_2	0,004	>0,150
Kolmogorov-Smirnov	x_3	0,004	>0,150
Kolmogorov-Smirnov	x_4	0,005	>0,150
Kolmogorov-Smirnov	x_5	0,003	>0,150
Kolmogorov-Smirnov	x_6	0,002	>0,150

After proving the suitability of all variables for univariate normal distribution, multivariate normal distribution in which all variables were evaluated together was examined. Multivariate normal distribution results evaluated with Henze-Zirkler's test and the Q-Q chart are shown in Table 6.

Table 6. Multivariate normal distribution test results

Test	Variable	P value	Normality
Henze-Zirkler	$x_1 \ldots . x_6$	0,4539	Yes

As seen in the table, six variables were found to be suitable for multivariate normal distribution.

- **No Autocorrelation Assumption:** This assumption was tested using the Box-Ljung statistic of the time independence of the variables. It was observed that there was no autocorrelation for all six variables. As a result, it has been determined that the Hotelling T^2 control chart is suitable for the assumptions.

For the measurement results of six variables on 26697 units, T^2 values were calculated using Equation (1) and UCL value was calculated using Equation (2). The Hotelling T^2 control chart created according to T^2 values is shown in Figure 5. The upper control limit of the control chart was found to be 30.1. It can be seen that the T^2 values of 25893 samples are between UCL and 0, while 804 samples are outside the UCL.

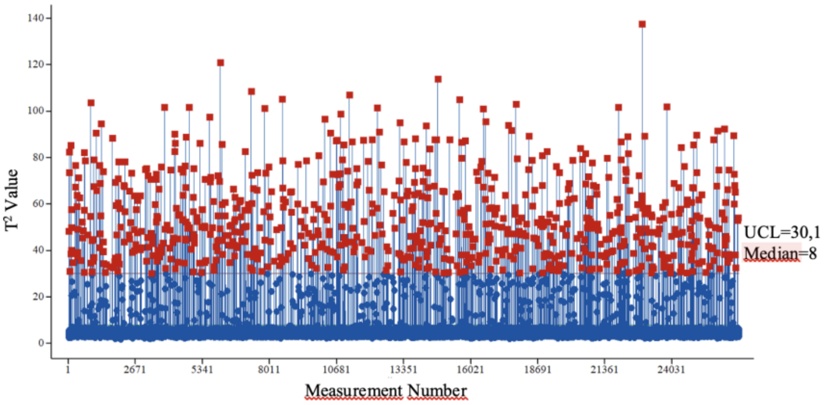


Figure 5. Hotelling T^2 Control Chart

5.3. Identification variables that cause uncontrolled situations with the MYT method

For each of the 804 samples that signaled that the process was out of control, the variable(s) causing the out-of-control situation were determined using Equations 4-6 of the MYT decomposition method. The results of MYT decomposition method implementation for six variables of 10 samples selected from 804 samples are as shown in Table 7.

"0" in the last column of the table represents variables that are under control, and "1" represents variables that cause an out-of-control situation. For example; variable class “100001”; It means that the out-of-control situation occurs due to the variables x_1 and x_6, while the other variables remain within the control limit according to the calculated threshold value and do not affect the out-of-control situation. Since there are 6 variables evaluated, there are $(2^6 - 1) = 63$ possible out-of-control situations (Niaki and Abbasi, 2005). The number of samples in which these possible out-of-control situations were observed in the examined data set is as shown in Table 8.

Table 7. MYT unconditional part T^2 values and out-of-control situations

Sample No	T_1^2	T_2^2	T_3^2	T_4^2	T_5^2	T_6^2	Condition
19	9,818	0,298	0,560	0,272	2,978	35,053	100001
27	1,141	36,305	9,005	25,855	1,093	0,099	011100
38	24,641	6,589	11,685	0,704	8,655	31,180	111011
52	17,048	0,883	6,350	0,333	0,787	5,257	101001
92	1,094	6,014	27,093	0,760	2,775	0,421	011000
95	0,252	1,093	0,486	12,159	33,232	38,175	000111
109	0,575	0,509	15,589	15,671	11,380	14,084	001111
110	6,452	0,848	10,892	0,354	13,798	16,987	101011
115	4,532	0,640	29,088	0,894	3,802	1,148	101000
175	0,686	0,735	10,161	21,790	0,879	1,049	001100

Table 8. Out-of-control situations and number of samples encountered

Condition	Number of Sample	Condition	Number of Sample	Condition	Number of Sample
000001	14	010110	5	101011	5
000010	4	010111	12	101100	10
000011	7	011000	24	101101	10
000100	17	011001	20	101110	5
000101	21	011010	5	101111	7
000110	10	011011	5	110000	12
000111	13	011100	11	110001	16
001000	18	011101	10	110010	7
001001	30	011110	11	110011	2
001010	8	011111	3	110100	16
001011	14	100000	8	110101	13
001100	30	100001	15	110110	5
001101	23	100010	12	110111	3
001110	11	100011	11	111000	12
001111	10	100100	100	111001	13
010000	8	100101	25	111010	3
010001	23	100110	5	111011	7
010010	10	100111	4	111100	13
010011	5	101000	13	111101	9
010100	16	101001	15	111110	8
010101	14	101010	8	111111	0

There is no situation in which all variables have an impact on the out-of-control situation, expressed by the "111111" variable class. For this reason, 62 different out-of-control situations will be considered in the data set. Variables x_1 and x_4 (100100) cause 100 of the 804 out of control situations, which is the most common situation, to be out of control. The least common out-of-control situations belong to the variable classes 011111, 110111 and 111010, with 3 samples each. Out-of-control situations will be called classes in the following sections of the study.

5.4. Development and Implementation Proposed Model Based on Ensemble Algorithm

5.4.1. Create a dataset

While the input data set consists of measurement values of the samples collected from the process, the output data set is the classes that express the variables that cause out-of-control situations obtained as a result of the calculations made in the previous steps. The number of samples for 62 classes varies between 3 and 100. This situation creates an unbalanced data set in terms of sample numbers between classes. Since real data was used in order not to affect the classification accuracy, synthetic data was produced with the help of the SMOTE method, using the highest number of samples as 100, to complete 100 samples for all classes. Thus, we continued with 6200 data belonging to 62 uncontrolled classes.

5.4.2. Implementation of single machine learning algorithms

When basic machine learning algorithms are used single, the parameters that provide the best classification performance are estimated heuristic, taking into account the preliminary information of the data set. The models were redesigned and trained according to each parameter and the results were obtained. A comparison of the success rates obtained from the algorithms was made by determining the appropriate parameter values. Cross-validation method was used for the training phase of the models established with classification algorithms. Cross-validation is a statistical method used to evaluate and compare learning algorithms by dividing data into two parts, one used to learn or train a model and the other used to validate the model (Refaeilzahed et al., 2009). In k-fold cross validation, the data is first divided into k equal sized partitions. Then, a selected partition test set is considered as the remaining k-1 partition training set. In the next phase, a different section is selected for testing and the remaining ones form the training set. The cluster to be selected does not have a priority or importance, each section is of equal importance. This process is repeated k times, each time with a different subsection test set, so that each section is used for both testing and training. In order to ensure consistency of the study, all models were trained using the same parameters. For the number of folds, the value "10", which is frequently used in studies (Refaeilzadeh et al., 2009; Zhang et al., 2019; Jonathan et al., 2019; Karimi et al., 2015; Ramezan et al., 2019; Yu and Feng, 2014.), was taken. Additionally, the sampling type was selected automatically and folding sampling was used because the result values were nominal. Multi-class performance criteria were used to evaluate the classification success rates obtained using the parameter values determined for all algorithms. Rapidminer Studio 9.6 Program was used in all analyses.

***DT algorithm*:** Some of the parameters used for the DT algorithm are shown in Table 9. Similar to previous studies for the splitting process in the tree (Dreiseitl et al., 2001; Anwar et al., 2014), the criterion for selecting the attributes was determined as information gain, which calculates the entropy and selects the least valuable one as the splitting criterion. The maximum depth value was selected as 20 by trying 31 values between 0-30. The confidence level was selected by performing 11 trials in 0.1 step increments between 0 and 1. For the values of other parameters, the program was run with default values.

Table 9. DT parameters

Parameter	Value
Criterion	Information gain
Maximum depth	20
Confidence level	0,1
Min. earnings	0,1

K-NN algorithm: The number of nearest neighbors (k) used for classification was determined as 3, which gives the highest performance, by trying odd numbers between 1-13, as shown in Table 10. Since the accuracy rate remained constant until k=9 and then started to decrease, k=3 was taken as the first highest value among 7 trials.

Table 10. Performance values according to K-NN k parameters

k	Accuracy Rate
1	86,53%
3	**88,85**%
5	88,85%
7	88,85%
9	88,85%
11	88,74%
13	86,53%

The measurement type parameter used to detect the nearest neighbors was chosen as numerical measurements since the data set contains numerical values and Euclidean distance because it is the most frequently used distance type (Hu et al., 2016). The parameters used for K-NN are shown in Table 11.

Table 11. K-NN algorithm parameters

Parameter	Value
K	3
Measurement type	Numerical Measures
Mixed Measure	Euclidean Distance

NB algorithm: Classification is made based on only one parameter, Laplace correlations, there are no other parameters (Anwar et al., 2014).

Multi-class support vector machine algorithm: Since the process discussed in the study is multi-class, the M-SVM algorithm was used. For classification, a one-versus-one approach of the multi-class support vector was used, which has proven successful in the work of Du et al. (2012). The type of kernel function was determined as a radial basis function, taking into account past studies (Du et al., 2012; Farhan et al., 2014; Lu et al., 2011; Onel et al., 2019) and the data set structure. Other parameters were run with the program's default values. The parameters of the M-SVM algorithm are as shown in Table 12.

Table 12. M-SVM algorithm

Parameter	Value
SVM approach	One-versus-one
Kernel Type	Radial basis function

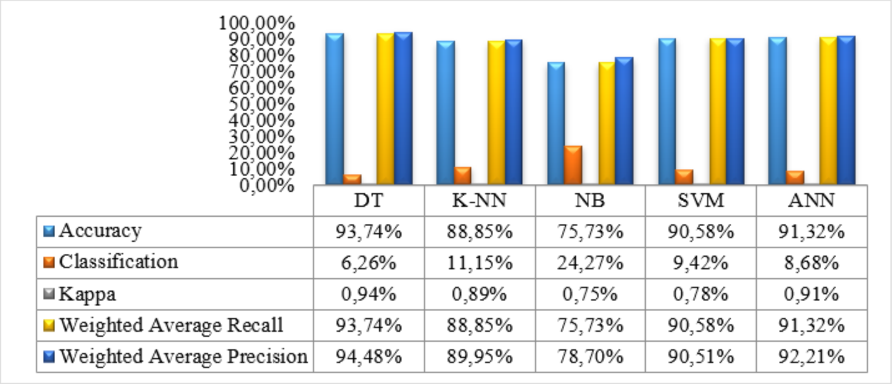
Artificial Neural Networks Algorithm: Feed-forward back-propagation multilayer perceptron neural network has been determined to be suitable from studies in the literature (Aparisi et al., 2006; Niaki and Abbasi, 2005; Salehi et al., 2012). In the network structure, there are input consisting of six variables, two hidden layers containing 100 neurons each, and 62 outputs consisting of classes. The parameters used for the neural network are shown in Table 13. As in classification and prediction studies, the activation function was used as sigmoid (Chen and Wang, 2004; Yu et al., 2009; Maleki and Amiri, 2015). In neural networks, the weight of each connection is updated to reduce the value of the error function. Using the training cycle parameter, the number of times this process should be repeated was tried 7 times, every 50 units in the range of 200-500, and was determined as 500. Learning rate and other parameters were used assuming default values (Shao and Lin, 2019).

Table 13. ANN algorithm parameters

Parameter	Value
Activation function	Sigmoid
Training Cycle	500
Learning rate	0,01

5.4.3. Performance evaluation of single machine learning algorithms

The performances of the five basic machine learning algorithms are shown in Figure 6. When the results are compared, it is seen that the DT algorithm is the most successful classification algorithm compared to the others. Thus, DT was determined as the basic classification algorithm.



	DT	K-NN	NB	SVM	ANN
■ Accuracy	93,74%	88,85%	75,73%	90,58%	91,32%
■ Classification	6,26%	11,15%	24,27%	9,42%	8,68%
■ Kappa	0,94%	0,89%	0,75%	0,78%	0,91%
■ Weighted Average Recall	93,74%	88,85%	75,73%	90,58%	91,32%
■ Weighted Average Precision	94,48%	89,95%	78,70%	90,51%	92,21%

Figure 6. Performance comparison of Single Machine Learning Algorithms

5.4.4. Combining of the selected machine learning algorithm with ensemble methods

The decision tree algorithm, which was selected with the highest classification success among single algorithms, was combined with bagging and boosting methods using previously determined parameters.

Combining of with the bagging ensemble method: The results obtained by combining the DT algorithm in parallel with 10 repetitions are shown in Table 11.

Combining of with the boosting ensemble method: The results obtained by combining the DT algorithm sequentially (with the Adaboost method) in 10 iterations are shown in Table 14.

Table 14. Ensemble methods performance values

Criterion	Adaboost	Bagging
Accuracy	95,08%	94,97%
Classification Error	4,92%	5,03%
Kappa	0,950%	0,949%
Weighted Average Sensitivity	95,08%	94,97%
Weighted Average Precision	95,56%	95,46%

When the results are examined, it is seen that combining the decision trees sequentially with the Adaboost method increases the accuracy.

5.4.5. Ensemble of Ensemble Model

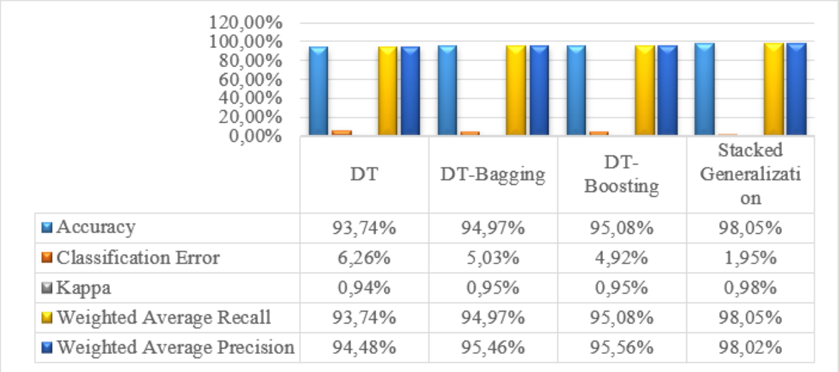
In the stacked generalization method, which has a different working principle from the two methods, different types of classification algorithms are combined sequentially. The model of the study is formed by combining DT-Bagging and DT-Adaboost ensemble algorithms. Performance values are shown in Table 15.

Table 15. Stacked generalization performance values

Criterion	Value
Accuracy	98,06 %
Classification Error	1,94 %
Kappa	0,980 %
Weighted Average Recall	98,06 %
Weighted Average Precision	98,27 %

5.4.6. Performance Evaluation of the Proposed Model

The classification performances obtained by combining the DT algorithm single, the ensemble algorithms sequentially and in parallel, and the last combination of the ensemble algorithms are shown in Figure 7. It is seen that the merging process gradually increases the performances. While the classification accuracy was 93.74% when using the DT algorithm alone, DT-bagging was 94.97%, DT-boost was 95.08%, and the accuracy performance of the model created with the stacked generalization method, which was seen as the most successful, was 98.06%. Thus, it can be seen that the developed model has the ability to classify with higher accuracy.



	DT	DT-Bagging	DT-Boosting	Stacked Generalization
■ Accuracy	93,74%	94,97%	95,08%	98,05%
■ Classification Error	6,26%	5,03%	4,92%	1,95%
■ Kappa	0,94%	0,95%	0,95%	0,98%
■ Weighted Average Recall	93,74%	94,97%	95,08%	98,05%
■ Weighted Average Precision	94,48%	95,46%	95,56%	98,02%

Figure 7. Comparison of the performance of the proposed model with other models

6. Discussion And Conclusion

In order for machine learning algorithms to make accurate predictions, their performance is required to be at the highest level. To achieve this, ensemble machine learning methods have been used. Ensemble algorithms are combined with the stacked generalization algorithm, which is an ensemble method that allows combining different algorithms. The single algorithm was improved by combining the Bagging and Boosting ensemble methods with the single algorithm, and then the two improved methods were combined. The intended target was achieved with the high success rates obtained as a result of the Implementation study carried out to determine the causes of uncontrolled situations in the casting process of the hydraulic pump cover. Thanks to the developed model, it will be possible to predict which variable is the cause in case the newly taken samples are out of control, without the need for multivariate control charts. Thus, faster and more accurate corrective measures can be taken. Great improvements in product quality can be achieved by applying corrective actions not on the product but during the production process.

The stacked generalization combination method used in the developed model has not been encountered before in the field of quality control or in a study on determining the causes of out-of-control situations. The limitation of the study is that only basic machine learning algorithms were used for single algorithm use. As future work, models will be enriched by using different single machine learning algorithms. It is thought that the algorithms will use an optimization technique instead of finding the parameters by trying them intuitively, and the model can be applied to different processes by changing the variables. In addition, accuracy will be evaluated by including feature selection in the study.

Acknowledgement: This study was produced from the doctoral thesis prepared by the first author under the supervision of the second author, which was carried out in the Department of Industrial Engineering, Institute of Natural Sciences, Sakarya University.

References

Agog, N. S., Dikko, H. G., Asiribo, O. E., 2014. Determining out-of-control variable(s) in a multivariate quality control chart. Sci. Africana, 13(2), 266–280.

Agrawal, R., Imielinski, T., 1993. Swami, A., mining association rules between sets of items in large databases. ACM SIGMOD, 1–10.

Ahsan, M., Mashuri, M., Lee, M. H., Kuswanto, H., Prastyo, D. D. 2020. Robust adaptive multivariate Hotelling's T^2 control chart based on kernel density estimation for intrusion detection system. Expert Systems with Applications, 145, 113105.

Alfaro, E., Alfaro, J.L., Gamez M., Garcia N., 2009. A boosting approach for understanding out-of-control signals in multivariate control charts. Int. J. Prod. Res., 47(24), 6821–6834.

Alpaydın, E., 2012. Yapay Öğrenme. 3. Edition. Boğaziçi University, 207-341.

Anwar, H., Qamar, U. Qureshi, A. W. M., 2014. Global optimization ensemble model for classification methods. Sci. World J., 1-9.

Aparisi, F., Avendaño, G., Sanz, J., 2006. Techniques to interpret T^2 control chart signals. IIE Trans., Institute Ind. Eng., 38(8), 647–657.

Asadi, A., Farjami Y., 2019. Online mean shift detection in multivariate quality control using boosted decision tree learning. J. Syst. Manag., vol. 2, 081–106.

Bersimis, S., Sgora, A., Psarakis, S. 2022. A robust meta-method for interpreting the out-of-control signal of multivariate control charts using artificial neural networks. Quality and Reliability Engineering International, 38(1), 30-63.

Bilgin, M., 2018. Veri Biliminde Makine Öğrenmesi Makine Öğrenmesi Teorisi ve Algoritmaları. 2. Edition Papatya Bilim, 31-138.

Blagus, R., Lusa, L., 2013. SMOTE for high-dimensional class-imbalanced data. BMC Bioinformatics, 14(16), 1471–2103.

Boullosa, D., Larrabe, J. L., Lopez, A., Gomez M. A., 2017. Monitoring through T^2 Hotelling of cylinder lubrication process of marine diesel engine. Appl. Therm. Eng., 110, 32–38.

Breiman, L. 1996. Bagging predictors. Machine learning, 24, 123-140.

Çetin, S., Birgören B., 2007. Çok değişkenli kalite kontrol çizelgelerinin döküm sanayiinde uygulanması. Gazi Üniv. Müh. Mim. Fak. Der., 22(4), 809–818.

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.

Chen. L. H., Wang T. Y., 2004. Artificial neural networks to classify mean shifts from multivariate $\chi 2$ chart signals. Comput. Ind. Eng., 47(2–3), 195–205.

Cheng, C. S., Cheng, H. P., 2008. Identifying the source of variance shifts in the multivariate process using neural networks and support vector machines. Expert Syst. Appl., 35(1–2),198–206.

Cheng, C.S., Lee H.T., 2012. Identifying the out-of-control variables of multivariate control chart using ensemble SVM classifiers. J. Chinese Inst. Ind. Eng., 29(5), 314–323.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn., 20(3), 273–297.

Das, N., Prakash, V., 2008. Interpreting the out-of-control signal in multivariate control chart — a comparative study. Int. J. Adv. Manuf. Technol., 37, 966–979.

Dreiseitl, S., Machado, O, L., Kittler, H., Vinterbo, S., Billhardt, H., Binder, M., 2001. A comparison of machine learning methods for the diagnosis of pigmented skin lesions. J. Biomed. Inform., 34(1), 28-36.

Du, S., Lv, J., Xi, L., 2012. On-line classifying process mean shifts in multivariate control charts based on multiclass support vector machines. Int. J. Prod. Res., 50(22), 6288–6310.

Farhan, S., Fahiem, M. A., Tauseef, H., 2014. An ensemble-of-classifiers based approach for early diagnosis of alzheimer's disease: Classification using structural features of brain images. Comput. Math., Methods Med., 2014.

Gowda, S., Kumar, H., Imran, M., 2018. Ensemble based learning with stacking. Boosting and Bagging for Unimodal Biometric Identification System, 30-36.

Guh, R. S., Shiue Y. R., 2008. An effective application of decision tree learning for on-line detection of mean shifts in multivariate control charts. Comput. Ind. Eng., 55(2), 475–493.

Han, J., Kamber, M., Pei, J., 2012. Data mining. concepts and techniques. The Morgan Kaufmann Series in Data Management Systems, 3. Edition.

Hawkins, D. M., 1991. Multivariate quality control based on regression-adiusted variables. Technometrics, 33(1), 61–75.

Hossin, M, Sulaiman, M., N, 2015. A review on evaluation metrics for data classification evaluations. Int. J. Data Min. Knowl. Manag. Process, 5(2), 01–11.

Hotelling H., Multivariable quality control—illustrated by the air testing of sample bombsight, McGraw Hill, 111-184, 1947.

Hu, L. Y., Huang, M. W., Ke, S. W., Tsai, C. F., 2016. The distance function effect on k-nearest neighbor classification for medical datasets. Springerplus, 5(1).

Huda, S., Abdollahian, M., Mammadov, M., Yearwood, J., Ahmed S., Sultan I., 2014. A hybrid wrapper-filter approach to detect the source(s) of out-of-control signals in multivariate manufacturing process. Eur. J. Oper. Res., 237(3), 857–870.

Jackson, J. E., 1985. Multivariate quality control. Commun. Stat. Theory Methods, 14(11), 2657–2688.

Jiang, J., Song, H.-M., 2017. Diagnosis of out-of-control signals in multivariate statistical process control based on bagging and decision tree. Asian Bus. Res., 2(2).

Jonathan, O., Omoregbe, N., Misra, S., 2019. Empirical comparison of cross-validation and test data on internet traffic classification methods. Journal of Physics: Conference Series, 1299(1), 1-9.

Joshi, K., Patil, B. 2022. Multivariate statistical process monitoring and control of machining process using principal component-based Hotelling T^2 charts: A machine vision approach. International Journal of Productivity and Quality Management, 35(1), 40-56.

Karimi, S., Yin, J., Baum, J., 2015. Evaluation methods for statistically dependent text. Comput. Linguist., 41(3), 539–548.

Lantz, B., 2013. Machine learning with R: learn how to use R to apply powerful machine learning methods and gain an insight into real-world applications. Birmingham: Packt Publishing Ltd; 66-343.

Li, J., Jin, J., Shi, J., 2008. Causation-based T^2 decomposition for multivariate process monitoring and diagnosis. J. Qual. Technol., 40 (1), 46–58.

Li, T., Hu, S., Wei, Z., Liao, Z., 2013. A framework for diagnosing the out-of-control signals in multivariate process using optimized support vector machines. Math. Probl. Eng., 2013(2), 1–9.

Lowry, C. A., Woodall, W. H., Champ, C. W., Rigdon, S. E., A multivariate exponentially weighted moving average control chart, Technometrics, 34(1), 46–53, 1992.

Lu, C. J., Shao, Y. E., Li, P. H., 2011. Mixture control chart patterns recognition using independent component analysis and support vector machine. Neurocomputing, 74(11), 1908-1914.

Maimon, L., Rokach, O., 2010. Data mining and knowledge discovery handbook. 2. Edition. Springer London, 165-174.

Maleki, M. R., Amiri, A., 2015. Simultaneous monitoring of multivariate-attribute process mean and variability using artificial neural networks. J. Qual. Eng. Prod. Optim., 1(1), 43–54.

Mason, R. L., Champ, C. W., Tracy, N. D., Wierda, S. J., & Young, J. C. (1997). Assessment of multivariate process control techniques. Journal of quality technology, 29(2), 140-143.

Mason, R. L., Tracy, N. D., Young, J. C., 1995. Decomposition of T^2 for multivariate control chart interpretation. J. Qual. Technol., 27(2), 99–108.

Mitchell, T. M., 2014. Machine learning. McGraw-Hill Science, 52-155.

Mohammed, M., Khan, M. B., Bashier, E. B. M., 2016. Machine learning: Algorithms and applications. 1. Edition. CRC Press, 5-11.

Montgomery D. C., 2009. Introduction to statistical quality control. 6. Edition. John Wiley & Sons, 499-507.

Niaki, S. T. A., Abbasi. B., 2005. Fault diagnosis in multivariate control charts using artificial neural networks. Qual. Reliab. Eng. Int., 21(8), 825–840.

Onan, A., 2018. Particle swarm optimization based stacking method with an application to text classification. Acad. Platf. J. Eng. Sci., 6(2), 134–141.

Onel, M., Kieslich, C. A., Pistikopoulos, E. N., 2019. A nonlinear support vector machine-based feature selection approach for fault detection and diagnosis: Application to the Tennessee Eastman process. AIChE J., 65(3), 992–1005.

Özel, S. 2005. Çok değişkenli kalite kontrolün döküm sanayiinde uygulanması, Master's Thesis, Kırıkkale University, YOK Thesis Center.

Öztemel E., 2003. Yapay Sinir Ağları. İstanbul, Papatya Yayınları, 7.

Parra, M. G., P. Loaiza, R., 2003. Application of the multivariate T^2 control chart and the Mason Tracy Young decomposition procedure to the study of the consistency of ımpurity profiles of drug substances. Qual. Eng., 16(1), 127–142.

Pei, X., Yamashita, Y., Yoshida, Matsumoto, M., S., 2006. Discriminant analysis and control chart for the fault detection and identification. Comput. Aided Chem. Eng.,21, 1281-1286.

Rakhmawan, S. A., Omar, M. H., Riaz, M., Abbas, N. 2023. Hotelling T^2 control chart for detecting changes in mortality models based on machine-learning decision tree. Mathematics, 11(3), 566.

Ramezan, C. A., Warner, T. A., Maxwell, A. E., 2019. Evaluation of sampling and cross-validation tuning strategies for regional-scale machine learning classification. Remote Sens., 11(185), 1-22.

Rao, O. R. M., Subbaiah, K.V., Rao, K. N., Rao T. S., 2013. Application of multivariate control chart for improvement in quality of hotmeal-a case study. Int. J. Qual. Res., 7(4), 623–640.

Refaeilzadeh, P., Tang, L., Liu, H., 2009. C Cross-validation. Springer, Boston, 1-3.

Robert J. C. Y., Mason L., 2002. Multivariate statistical process control with industrial applications. Society for Industrial and Applied Mathematics, 5-17.

Rokach, L., 2010. Ensemble-based classifiers. Artif. Intell. Rev., 33(1–2), 1–39.

Sabahno, H., Amiri, A. 2023. New statistical and machine learning based control charts with variable parameters for monitoring generalized linear model profiles. Computers & Industrial Engineering, 184, 109562.

Salehi, M., Kazemzadeh, R. B., Salmasnia, A., 2012. On line detection of mean and variance shift using neural networks and support vector machine in multivariate processes. Appl. Soft Comput. J., 12(9), 2973–2984.

Shao, Y. E., Lin, S. C., 2019. Using a time delay neural network approach to diagnose the out-of-control signals for a multivariate normal process with variance shifts. Mathematics, 7(10).

Şişci, M., Torkul, Y. E., Selvi, İ. H. 2022. Machine learning as a tool for achieving digital transformation. Knowledge Management and Digital Transformation Power, 55.

Song, H., Xu, Q., Yang, H., Fang, J., 2017. Interpreting out-of-control signals using instance-based bayesian classifier in multivariate statistical process control. Commun. Stat. Simul. Comput., 46(1).

The Royal Society, 2017. Machine learning: the power and promise of computers that learn by example, 5-6.

Ulen, M., Demir, I., 2013. Application of multivariate statistical quality control in pharmaceutical industry. Balk. J. Math.,1, 93–105.

Utgoff, P. E. Berkman, N. C., Clouse, J. A., 1997. Decision Tree Induction Based on Efficient Tree Restructuring. Kluwer Academic Publishers, 29, 5-44.

Woodall W. H., Ncube M. M., Multivariate CUSUM quality-control procedures, technometrics, 27(3), 285–292, 1985.

Yadav, M., Yadav, A., Kumar N., 2015. An introduction to neural network methods for differential equations. Springer.

Yang, W. A., 2015. Monitoring and diagnosing of mean shifts in multivariate manufacturing processes using two-level selective ensemble of learning vector quantization neural networks. J. Intell. Manuf., 26(4), 769–783.

Yılmaz, H., 2012. Çok değişkenli istatistiksel süreç kontrolü: Bir hastane uygulaması, Master's Thesis, İstanbul Teknik University, YOK Thesis Center.

Yu, J. Bo., Xi, L. Feng., 2009. A neural network ensemble-based model for on-line monitoring and diagnosis of out-of-control signals in multivariate manufacturing processes. Expert Syst. Appl., 36(1), 909–921.

Yu, Y., Feng, Y., 2014. Modified cross-validation for penalized high-dimensional linear regression models. J. Comput. Graph. Stat., 23(4), 1009-1027.

Zhang, Y., Li, M., Han, S., Ren, Q., Shi, J., 2019. Intelligent identification for rock-mineral microscopic images using ensemble machine learning algorithms. Sensors, 19(9), 1-14.

Zhang, Y., Ma, C., 2012. Ensemble machine learning. Springer US.

Zhou, Z. H., 2012. Ensemble methods: foundations and algorithms Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, Taylor & Francis.

Research Article
Journal of Intelligent Systems: Theory and Applications 7(2) (2024) 145-158
DOI: 10.38016/jista.1440879

Leveraging Machine Learning Methods for Predicting Employee Turnover Within the Framework of Human Resources Analytics

Zeynep Taner[1], Ouranıa Areta Hıziroğlu[2*], Kadir Hıziroğlu[3]

[1, 2, 3] Department of Management Information Systems, İzmir Bakırçay University, İzmir, Türkiye

zeynepyt78@gmail.com, ourania.areta@bakircay.edu.tr, kadir.hiziroglu@bakircay.edu.tr

Abstract

Employee turnover is a critical challenge for organizations, leading to significant costs and disruptions. This study aims to leverage Machine Learning (ML) techniques within the framework of Human Resources Analytics (HRA) to predict employee turnover effectively. The research evaluates and compares the performance of six widely used models: Decision Trees, Support Vector Machines (SVM), Logistic Regression, Random Forest, XGBoost, and Artificial Neural Networks. These models were implemented using the R programming language on an open-source dataset from IBM. The methodology involved data preprocessing, splitting into training, validation and testing sets, model training, and performance evaluation using metrics such as accuracy, sensitivity, specificity, precision, F1-score, and ROC-AUC. The results indicate that the Logistic Regression model outperformed the other models, achieving high accuracy and a good F1-score. The study concludes by emphasizing the importance of HRA and ML techniques in predicting and managing employee turnover, while discussing limitations such as class imbalance and the need for more rigorous performance evaluation. Future research directions include exploring alternative models, feature selection techniques, and addressing class imbalance.

Keywords: Human resources analytics, Employee turnover prediction, Machine learning models.

Makine Öğrenimi Yöntemlerini İnsan Kaynakları Analitiği Çerçevesinde İşten Ayrılma Tahminleri için Kullanma

Öz

Çalışan devir oranı, kuruluşlar için önemli bir zorluk oluşturmakta ve önemli maliyetlere ve aksaklıklara yol açmaktadır. Bu çalışma, insan kaynakları analitiği çerçevesinde makine öğrenimi tekniklerini etkin bir şekilde kullanarak çalışan devirini öngörmeyi amaçlamaktadır. Araştırma, altı yaygın olarak kullanılan modelin performansını değerlendirmekte ve karşılaştırmaktadır: Karar Ağaçları, Destek Vektör Makineleri, Lojistik Regresyon, Rastgele Orman, XGBoost ve Yapay Sinir Ağları. Bu modeller, IBM'den açık kaynaklı bir veri kümesi üzerinde R programlama dili kullanılarak uygulanmıştır. Çalışmanın metodlolojisi, veri ön işleme, eğitim, doğrulama ve test setlerine bölme, model eğitimi ve doğruluk, hassasiyet, özgünlük, hassasiyet, F1-skoru ve ROC-AUC gibi ölçümleri kullanarak performans değerlendirmeyi içermektedir Sonuçlar, Lojistik Regresyon modelinin diğer modellerden daha iyi bir performans sergilediğini, yüksek doğruluk ve iyi bir F1-skoru elde ettiğini göstermektedir. Çalışma kasapmında, çalışan devir oranını öngörmek ve yönetmek için insan kaynakları analitiği ve makine öğrenmesi tekniklerinin önemi vurgulanarak, sınıf dengesizliği gibi sınırlamaları ve daha güvenilir performans değerlendirmesi gereksinimine yönellik tartışmalara da yer vermektedir. Çalışmanın son kısmında, gelecek araştırma konuları çerçevesinde alternatif modellerin keşfedilmesi, özellik seçim teknikleri kullanılarak sonuçların değerlendirilmesi ve sınıf dengesizliğini gidermeye dönük hususlar ele alınmaktadır.

Anahtar Kelimeler: İnsan kaynakları analitiği, Çalışan devir hızı tahmini, Makine öğrenimi modelleri.

* Corresponding Author
E-Mail: ourania.areta@bakircay.edu.tr

Recieved : 21 Feb 2024
Revision : 6 May 2024
Accepted : 9 Jul 2024

1. Introduction

Human Resource Management (HRM) has undergone transformations to cope with ongoing technological advancements and dynamic business requirements. One such transformation is the adoption of HRA, which involves analyzing HR data on a larger scale to support evidence-based decision-making related to human performance, satisfaction, engagement, and ultimately, turnover. HRA has become increasingly important in understanding various processes that contribute to overall business success and competitive advantage (Van Vulpen, 2023).

The suitability of leveraging ML techniques for analyzing employee turnover within the HRA framework lies in their ability to identify complex patterns and relationships in large datasets, which may not be apparent through traditional statistical methods. MLmodels can learn from historical data and provide accurate predictions, enabling organizations to proactively identify employees at risk of turnover and take appropriate measures to retain valuable talent.

A critical aspect of HRA is the prediction of employee turnover, as high turnover rates can incur significant costs and impact productivity (Yavuz, 2016). Numerous studies have examined employee turnover and its reasons, highlighting the importance of retaining and rewarding the best employees (Aarons et al., 2009; Peryön, 2017, 2018; Randstad, 2022, 2023; Gallup, 2024). Effectively predicting employee turnover probabilities helps businesses improve workforce planning, reduce costs, and increase overall employee satisfaction (Moturi et al., 2023).

To address this challenge, the use of ML techniques within the framework of HRA has gained significant attention in recent years (Avrahami et al., 2022; Wijaya et al., 2021; Choi et al., 2021; Gao et al., 2019; Alsaadi et al., 2022). ML models can effectively predict employee turnover by learning from historical data and identifying patterns and relationships that may not be apparent through traditional statistical methods.

This study aims to evaluate and compare the performance of six widely used ML models - Random Forest, Logistic Regression, Artificial Neural Networks, Support Vector Machines, XGBoost and Decision Trees - in predicting employee turnover within the context of HRA. The choice of these models for predicting employee turnover in this study was based on their popularity, proven performance, and diversity of approaches (Breiman, 2001; Cortes & Vapnik, 1995; Friedman, 2001). These models represent a range of techniques, including tree-based methods, probabilistic models, and neural networks, capable of capturing complex relationships in the data (Demir & Çalık, 2021; Uzak, 2022). Some models, such as Decision Trees and Logistic Regression, offer interpretable results (Demir & Çalık, 2021), while others, like Random Forest and XGBoost, are known for their scalability and robustness to outliers and noise (Breiman, 2001; Friedman, 2001). The inclusion of simpler models allows for a comparison with more complex ones, assessing the trade-off between complexity and predictive performance (Liao, 2023). Moreover, these models have been successfully applied in previous studies on employee turnover prediction, providing evidence of their effectiveness in this context (Jain et al., 2020; Stachová et al., 2021).

Within the framework of its aim, the following research objectives were set:

- Evaluate and compare the performance of the trained models in predicting employee turnover using various metrics, including accuracy, sensitivity, specificity, precision, F1-score, and ROC-AUC (Receiver Operating Characteristic - Area Under the Curve).
- Identify the most effective model for predicting employee turnover and discuss the implications and limitations of the study.
- Provide recommendations for businesses and researchers to leverage ML techniques for effective employee turnover prediction and management.

By addressing these objectives, this study contributes to the existing body of knowledge in HRA and employee turnover prediction, while also providing practical insights for businesses to implement data-driven strategies for workforce management.

The study initially includes a literature review section, covering previous research around employee turnover prediction. This section identifies gaps in the existing literature and the contributions of this study. In the methodological part, elements such as the data set description, data preprocessing, model selection and training, model performance and evaluation are presented according to the research methodology. The following section presents the study's findings and the performance of the models, determining the best-performing model based on comparisons, offering also recommendations for usability of the tools for employee turnover prediction based on their suitability. Finally, the authors present the summary of the outcomes and future directions and recommendations in the conclusions section.

2. Literature Review

Human Resources Analytics is the process of collecting, analyzing, and making more effective decisions through insights derived from human resource data. HRA involves analyzing data from various sources within the enterprise using different methods to answer the right questions (Van Vulpen, 2023). Decision-making based on data enables organizations to gain a competitive advantage through more strategic and informed HRM (Shrivastava, Nagdev, and Rajesh, 2017).

In this study, while emphasizing the importance of data-driven decision-making in HRA, it also focuses on the analysis of employee turnover prediction within HRA applications. Employee turnover prediction analysis is a data analytics application that enables a business to predict employee departures in advance. This analysis has become a significant topic for businesses in recent years, providing important insights into workforce management and employee retention for employers and researchers (Wijaya et al., 2021; Ye et al., 2019; Liu & Liu, 2021; Schlechter et al., 2016; Putri & Rachmawati, 2022; Liao, 2023; Judrups et al., 2021; Chaudhary, 2022). Such studies help reduce workforce costs, increase employee satisfaction and productivity, and also aid in strategic human resources planning. In conducting employee turnover prediction analysis, the concept of "workforce turnover" comes into play, which refers to the number of employees leaving a business in a given period for various reasons, including voluntary and involuntary departures (Roche et al., 2015; Russell et al., 2017; Scanlan et al., 2013; Chisholm et al., 2011; Woltmann et al., 2008; Bogaert et al., 2019; Chapman et al., 2022; Roche et al., 2021; Poku et al., 2022; Onnis, 2017; Bardoel et al., 2020; Mayson & Bardoel, 2021; Healy & Oltedal, 2010; Russell et al., 2012; Belbin et al., 2012; Ashworth, 2006). High workforce turnover incurs significant costs, affecting training, recruitment, separation costs, and productivity (Yavuz, 2016). Therefore, having a model that can accurately predict the likelihood of employee departures is of great importance.

Empirical studies conducted within the scope of data analytics for predicting employee turnover have been presented in Table 1. The literature review table includes various research studies that have utilized different MLmodels and techniques for the purpose of predicting employee turnover. Each study is aimed at reducing the likelihood of employee turnover, targeting specific sectors and objectives. The studies vary in terms of features included, data sources, models and methods used, development tools, and evaluation metrics. Most research has two main objectives: "Increasing productivity" and "Reducing costs." For instance, a study using the K-nearest neighbours algorithm (Balcıoğlu & Artar, 2022) aims to increase efficiency, while a study on ML model selection for employee loss prediction in the telecommunications sector (Uzak, 2022) aims to reduce costs. Regarding data sources and size, some studies use open-source datasets, while others use in-house data, with data sizes ranging from small-scale studies to large datasets. The variables used include demographic (related to personal attributes of employees) and job-related variables (pertaining to employees' work experience and performance).

For development tools, programming languages such as Python or R have been used. Each study employed different ML models and methods, including Random Forest, Logistic Regression, K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree, Support Vector Machines (SVM), etc., in an attempt to predict employee turnover. According to the results of these studies, when examining the effectiveness of different MLmethods in predicting employee turnover, the studies "Prediction of Employee Turnover Probability with Machine Learning: K-Nearest Neighbors Algorithm (Balcıoğlu & Artar, 2022)" and "Employee Attrition Prediction (Yedida et al., 2018)" achieved high accuracy using the KNN algorithm. These results indicate KNN as an effective option for predicting employee turnover. Similarly, the studies "MLModel Selection for Employee Loss Prediction in the Telecommunications Sector (Uzak, 2022)" and "Predictive Analysis on the Example of Employee Turnover (Maisuradze, 2017)" have shown high accuracy with the Random Forest (RF) model, suggesting RF as a highly effective model for turnover prediction. The study "Prediction of Employee Turnover using ML(Shanthakumara et al., 2022)" used Artificial Neural Networks (ANN), showing ANN as a viable alternative for turnover prediction. The "Employee Attrition Prediction" study utilized Logistic Regression, indicating its effectiveness in turnover prediction. Metrics such as accuracy, precision, recall, and F1-score were commonly evaluated.

Additionally, some studies have utilized specific metrics like AUC (Area Under Curve). Data attributes in these studies include various factors such as demographic information (age, gender, education) and job-related information (position, salary, job satisfaction). These attributes have been used to predict the likelihood of employees leaving their jobs.

Despite the existing research, several limitations and gaps warrant further investigation:

- Limited comparative studies: While individual studies have explored the performance of specific ML models, there is a lack of comprehensive comparative analyses evaluating the effectiveness of different models on the same dataset.
- Inconsistent results: The existing literature presents inconsistent results regarding the most effective ML model for employee turnover prediction, suggesting that the choice of model may be context-dependent or influenced by factors such as data quality, preprocessing techniques, and feature selection.
- Lack of generalizability: Many studies have focused on specific industries or contexts, which may limit the generalizability of their findings to other organizational settings.
- Limited discussion of practical implications: While the studies demonstrate the potential of ML techniques for employee turnover prediction, there is often a lack of discussion regarding the practical implications and implementation challenges for businesses.
- Absence of rigorous model evaluation: Some studies have relied primarily on accuracy as the sole performance metric, overlooking the importance of other relevant metrics such as accuracy, sensitivity,

specificity, precision, F1-score, and ROC-AUC, which can provide a more comprehensive understanding of model performance.

This study aims to address these limitations by conducting a comprehensive comparative analysis of six widely used ML models (Random Forest, Logistic Regression, Artificial Neural Networks, Support Vector Machines, XGBoost and Decision Tree) on a publicly available dataset, evaluating their performance using multiple metrics, and discussing the practical implications and future research directions.

Table 1. Table of Studies Conducted in the Field of Human Resources Analytics

Title and Year	Purpose/ Objective	Attributes	Data Source/ Size	Development Tool	Model-Method and Techniques	Metric

Predicting Employee Attrition Using Machine Learning: A K-Nearest Neighbors Algorithm Approach (Balcıoğlu & Artar, 2022)	Increase efficiency	Demographic data; Age, Marital Status, Education level; Job-related data; Working hours, Position, Job satisfaction, Salary, Work arrangement	Open source - 1205	MATLAB R2020b	KNN (k=4) - %93
					KNN(K=1)
					KNN(K=6)
					KNN(K=8)
						Accuracy
						Precision
						Recall
						F1-Score

MLModel Selection for Predicting Employee Turnover in the Telecommunications Industry (Uzak, 2022)	Reduce costs	Demographic data; ID, Age, Gender, Marital status, Location, Child number, Military Service, School Type; Job-related data; Title, Function, Reason for Leaving, Status/Objective, Active/Inactive	Company data – 16655	Python	RF - %92,2 Logistic Regression - KNN - DVM – CART - Gradient Boosting Machine- YSA - XGBoost	Accuracy Precision Sensitivity F1-Score EAKA
Prediction of Employee Turnover using Machine Learning (Shanthakumara et al., 2022)	Increase efficiency	Demographic data; Age, Sex, Education; Job-related data; Position, Department, Salary, Overtime, Average Monthly Hours, Tenure, Number of Projects, Satisfaction, Work Accident	N/A – 15400	R	RF - %93 Naive Bayes - Logistic Regression	Accuracy
Employee Attrition Prediction (Yedida et al., 2018)	Increase efficiency	Job-related data; Average Monthly Hours, Number of Projects, Promotion in the Last Five Years, Seniority	Open source – 14999	Python	KNN - %94,32 Naive Bayes - Logistic Regression - MLP Classifier	AUC Accuracy F1-Score
Predicting the Perceived Employee Tendency of Leaving an Organization Using SVM and Naive Bayes Techniques (Emmanuel-Okereke & Anigbogu, 2022)	Reduce costs	Demographic data; Gender, Experience, Seniority, Education; Job-related data: Date of Entry, Job Safety, Working Hours, Job Satisfaction, Status/Objective	Survey - 514	Python	Naive Bayes - %100 DVM – RF – Decision Tree	Precision Recall F1-Score
Employee Turnover Prediction Using MLBased Methods (Kışaoğlu, 2014)	Reduce costs - Increase efficiency	Demographic data; Age, Race Job-related data; Performance, Job Satisfaction Survey Results, Job Transition/Change Networks, Status/Objective - "Will Leave", "Will Not Leave"	Open source - 25000	WEKA	DVM - Karar Ağacı - Naïve Bayes	Accuracy Precision Recall F1-Score
Employee Turnover Probability Prediction (Barın, 2022)	Reduce costs	Demographic data; Age, Seniority, Gender, Marital Status, Number of Children, Education; Job-related data:	Company data – 3282	R	Hierarchical Model - 69.4% Naive Bayes - RF	ROC-AUC

		Performance Score, Appreciation Score, Salary, Salary Increase, Promotion, First Year Information, Foreign Language; Status/Objective - Employed/Left				
Optimization of employee turnover through predictive analysis (Stachová, Baroková & Stacho, 2021)	Reduce costs	Demographic data; Age, Sex, Education, Marital status, Seniority; Job-related data; Business Travel, Position, Department, Commute Distance, Work-Life Balance, Hourly Wage, Monthly Income, Overtime, Working Hours, Salary, Salary Increase, Promotion	Open source – 1470	Python	RF -%87 Logistic Regression – Decision Tree - K-Means	Accuracy
Employee Churn Prediction using Logistic Regression and Support Vector Machine (Maharjan, 2021)	Reduce costs	Demographic data; Age, Education, Gender, Seniority, Marital Status; Job-related data: Position, Monthly Income, Job Satisfaction, Overtime, Performance, Training Duration (last year), Work-Life Balance, Work Experience, etc., Status/Objective, Employed/Left	Open source – 23436	Python	DVM -%84 Logistic Regression	Precision Recall F1-Score ROC-AUC Accuracy
Explaining and predicting employees' attrition: a MLapproach (Jain, Jain & Pamula, 2020)	Reduce costs	Job-related data; Satisfaction Level, Performance, Number of Projects, Average Monthly Hours, Work Accident, Promotion - Last 5 Years, Salary, Domain, Target Variable, Department Names (Sales, HR, Technical, Support, etc.)	Open source - 14.000+	Python	RF -%99 YSA - Decision Tree - Naive Bayes – Logistic Regression – DVM	Precision F1-Score Recall
Predictive Alaysis on the Example of Employee Turnover (Maisuradze, 2017)	Reduce costs	Demographic data; Age, Gender, Education, Seniority, Marital Status; Job-related data: Overtime, Job Satisfaction, Monthly Income, Performance, Distance from Home, Promotion, Work-Life Balance, Salary Increase, Position, Department	Open source - 1471	Python	RF- 98.62% DVM – YSA	ROC-AUC Accuracy
Employee turnover prediction and retention policies design: a case study (Ribes, Touahri & Perthame, 2017)	Reduce costs	Demographic data; Age, Experience, Gender, Ethnic Background, Education; Job-related data: Performance, Role Salary, Working Conditions, Job Satisfaction, Burnout, Seniority, Status/Objective,	Open source – 1000	R	Linear Discriminant-%75 DVM – KNN – RF -Naïve Bayes	Accuracy ROC-AUC

		Employed/Left				
Leveraging MLMethods for Predicting Employee Turnover Within the Framework of Human Resources Analytics (Current/Our Study)	Reduce costs	Demographic data; Age, Education, Gender, Seniority, Marital Status; Job-related data: Position, Monthly Income, Job Satisfaction, Overtime, Performance, Training Duration (last year), Work-Life Balance, Work Experience, etc., Status/Objective, Employed/Left	Open source - 1470	R	RF - YSA - Decision Tree - XGBoost – Logistic Regression – DVM	Precision Recall F1-Score ROC-AUC Accuracy

The following section presents the methodology that was employed so that the authors could meet the objectives of this study.

3. Research Methodology

The purpose of the research is to determine the most suitable and effective model for predicting employee turnover and to evaluate the performance of this model. The following sections describe the several stages that the authors undertook to meet the aim and objectives of this paper.

3.1 Data Source and Preprocessing

The dataset in question is from Kaggle platform, created by IBM data scientists and titled "IBM HR Analytics Employee Attrition & Performance" (Pavansubhash, 2016). The dataset comprises a total of 1470 employee records, (1233 employees and 237 leavers) with 35 features, including 34 independent variables and 1 dependent variable (Attrition). The independent variables encompass demographic information, job-related data, and other relevant factors, while the dependent variable is a binary indicator of employee attrition.

Data preprocessing involved removing variables with low analytical value, such as "EmployeeNumber," "EmployeeCount," "Over18," and "StandardHours." The remaining variables were then normalized for scaling to enable analysis that is more meaningful. Table 1 represents the preprocessed dataset and includes the types of variables in the dataset and their descriptions.

The preprocessed dataset was split then into training (60%), validation and testing (20% each) sets for models' development and evaluation where partitioning was carried out for each model separately.

Table 2. Preprocessed Data set

Order	Variable	Definition	Variable Type
	Demographic – Independent variable		
1	Age	Employee's Age	Numeric
2	Marital status	Marital Status (Single, Married, Divorced)	Categorical
3	Gender	Gender	Categorical
4	Education	Education Level (1: Below University, 2: University, 3: Bachelor's, 4: Master's, 5: Doctorate)	Numeric
5	Travel Status	Business Travel Frequency (No Travel, Rare Travel, Frequent Travel)	Categorical
	Job-related - Independent Variable		
1	Daily Wage	The amount of money a company is obligated to pay an employee for a day's work.	Numeric
2	Department	Department (Research and Development, Sales, Human Resources)	Categorical
3	Commute Distance	Distance between home and company	Numeric
4	Field of Study	Field of Education (Science, Medicine, Human Resources, Technical Degree, Marketing, Other)	Categorical
5	Environmental Satisfaction	Environmental Satisfaction Score (1: Low, 2: Medium, 3: High, 4: Very High)	Numeric
6	Engagement Level	Level of Job Involvement (1: Low, 2: Medium, 3: High, 4: Very High)	Numeric
7	Work-Family	Job Level (1 - 5)	Numeric
8	Role	Job Role (Sales Manager, Human Resources Manager, etc.)	Categorical
9	Job Satisfaction	Job Satisfaction (Low, Medium, High, Very High)	Numeric
10	Monthly Income	Employee's Monthly Income	Numeric
11	Salary Raise	Percentage of Salary Increase	Numeric
12	Number of Companies	Total number of companies the employee has worked for before	Numeric

	Worked At		
13	Job Satisfaction	Job Satisfaction (Low, Medium, High, Very High)	Numeric
14	Overtime	Employee's Overtime Status (Yes, No)	Categorical
15	Salary Raise %	Percentage of Salary Increase	Numeric
16	Performance Rating	Level of Performance Appraisal (Low, Good, Excellent, Outstanding)	Numeric
17	Communication Satisfaction	Level of Relationship Satisfaction (Low, Medium, High, Very High)	Numeric
18	Working Hours	Standard Working Hours	Numeric
19	Stock Option Level	Employee's Stock Option Level (0 - 3)	Numeric
20	Work Experience	Total Years of Working	Numeric
21	Training Duration (Last Year)	Training Duration Last Year	Numeric
22	Work-Life Balance	Work-Life Balance Level (1: Poor, 2: Good, 3: Better, 4: Best)	Numeric
23	Seniority	Years at the Company	Numeric
24	Tenure in Role	Years in Current Role	Numeric
25	Years with Current Manager	Years with Current Manager	Numeric
	Dependent Variable		
1	Attrition Status	Employee Attrition (Yes, No)	Categorical

3.2. Model Selection and Training

Six widely used ML models were selected for this study, namely as Random Forest, Logistic Regression, Artificial Neural Networks, Support Vector Machines, XGBoost and Decision Tree:

- Random Forest, an ensemble learning method, is known for its robustness, ability to handle large datasets with many features, and its effectiveness in both classification and regression tasks (Breiman, 2001).
- Logistic Regression, a classical statistical method, is often used when the dependent variable is categorical and provides interpretable results (Demir & Çalık, 2021).
- Artificial Neural Networks, inspired by the structure and function of biological neural networks, are capable of learning complex non-linear relationships between input features and the target variable (Demir & Çalık, 2021; Uzak, 2022).
- Support Vector Machines, a non-probabilistic binary linear classifier, are known for their ability to handle high-dimensional data and their effectiveness in both linear and non-linear classification tasks (Cortes & Vapnik, 1995).
- XGBoost, an ensemble learning method that combines multiple weak learners (decision trees) to create a strong learner, is known for its ability to handle complex interactions among features and its effectiveness in both classification and regression tasks (Friedman, 2001).
- Decision Trees, a simple yet powerful supervised learning algorithm, is known for their interpretability, ability to handle both categorical and numerical data, and effectiveness in capturing non-linear relationships between features and the target variable. They repeatedly divide the feature space into subsets based on the most informative features, creating a tree-like model that can be easily visualized and understood (Rokach & Maimon, 2005).

The researchers implemented the models using the R programming language. The installation, training, and performance evaluation of each model was carried out on the original dataset. The training process involved fitting each model to the training dataset, with 5-fold cross-validation to ensure the robustness and generalizability of the results. Cross-validation helps to assess the model's performance on different subsets of the data, reducing the risk of overfitting and providing a more reliable estimate of the model's performance on unseen data.

Training involved using the specified models with utilized to compare the models based on specific metrics (see part 3.3). The training set is the data used by the ML algorithm during its learning process. This dataset includes the input and output values for each example. The learning algorithm uses the data in the training set to learn the correct outputs for the inputs. For example, in text classification studies, the content of the input texts and the output categories are included in the training set. In contrast, the test set is used to validate and assess the performance of the trained model. The test dataset comprises data that are distinct and previously unseen in comparison to the training set. The model, trained during the learning process, makes predictions for the inputs in the test set. To evaluate the model's accuracy and performance, these predictions are compared with the actual outputs of the test data (Kutlugün et al., 2017).

More specifically, and with regards to each one of the ML models, the setup and evaluation took place as followed:

- Random Forest: A model containing 500 trees was established with the RandomForest package, and the classification performance of the model was examined in detail.

- Logistic Regression: Within the framework of the generalized linear model, a logistic regression model was created using the glm() function, and probability predictions were made.
- Artificial Neural Networks: A 10-neuron neural network model was established with the nnet package, and the classification predictions of the model were evaluated.
- Support Vector Machines (SVM): On the data divided into training and test sets, the SVM model was established by determining the optimal gamma and cost values through the e1071 package, and the classification performance was evaluated.
- XGBoost: The xgboost package was used, and various hyperparameters were adjusted with the train() function. These parameters include the maximum depth of trees (max_depth), learning rate (eta), and editing parameters (gamma). Additionally, optimal parameter combinations were determined using a comprehensive grid search method to further optimize the model.
- Decision Trees: A model to predict attrition was created using the Tree library, trained, and visualized by adding information to its branches. The accuracy of the model was evaluated on the test dataset with confusionMatrix.

3.3. Performance and Evaluation

The trained models were evaluated on the test dataset using various performance metrics, including accuracy, sensitivity, specificity, precision, F1-score, and ROC-AUC. These metrics provide a comprehensive assessment of the models' predictive capabilities, considering factors such as correct classifications, false positives, and false negatives. For the calculation of each of the aforementioned metrics, the following need to be defined:

- True Positives (TP): The number of instances that are actually positive and correctly predicted as positive by the model.
- True Negatives (TN): The number of instances that are actually negative and correctly predicted as negative by the model.
- False Positives (FP): The number of instances that are actually negative but incorrectly predicted as positive by the model.
- False Negatives (FN): The number of instances that are actually positive but incorrectly predicted as negative by the model.

Then, the metrics can be defined and calculated as followed:

- Accuracy: The proportion of correctly classified instances out of the total instances.
  Accuracy = (TP + TN) / (TP + TN + FP + FN)
- Sensitivity (Recall or True Positive Rate): The proportion of true positive predictions among all actual positive instances.
  Sensitivity = TP / (TP + FN)
- Specificity: The proportion of true negative predictions among all actual negative instances.
  Specificity = TN / (TN + FP)
- Precision: The proportion of true positive predictions among all positive predictions.
  Precision = TP / (TP + FP)
- F1-score: The harmonic mean of precision and recall, providing a balanced measure of the model's performance.
  F1-score = 2 * (Precision * Recall) / (Precision + Recall)
  The F1-score ranges from 0 to 1, with 1 being the best value and 0 being the worst.
- ROC: An aggregate measure of the model's performance, considering both its ability to identify positive instances (employee retention) and negative instances (employee turnover). It is calculated as the sum of the True Positive Rate (TPR) and the True Negative Rate (TNR) divided by 2. TPR measures the proportion of actual positive instances that are correctly identified, while TNR measures the proportion of actual negative instances that are correctly identified. The AUC (Area Under Curve) value measures the probability of the model correctly classifying a randomly selected positive example into a randomly selected negative example. The closer the AUC value is to 1, the better the model performs.

Within this framework, confusion matrices for each model have been included and explained in detail, providing insights into the models' performance in terms of true positives, true negatives, false positives, and false negatives. The confusion matrix is used to understand the model's performance more deeply and to examine the classification results in more detail. It is very valuable for determining which classes the model predicts better or worse, and which classes are associated with false positives or false negatives.

With the methodology clearly defined, next section presents the results obtained from the MLmodel testing and evaluation. The results and discussion section will analyze the performance of the selected models and interpret the findings.

4. Results and Discussion

4.1 Confusion Matrices

The confusion matrices provide a detailed breakdown of the models' performance in terms of TP, TN, FP, and FN. They help in understanding how well each model classified the instances into the correct categories. In the case of employee churn problem of this study, the positive class represents the employees who have not left the company whereas the negative class is the ones who left the company. Therefore, in the confusion matrices that will be provided below, indications with “yes” represent the negative classes (employee turnover) and with “no” refer to the positive classes (employee retention).

The confusion matrices for each of the ML models are as follows:

a. Random Forest:

- TN: The model correctly predicted 9 instances as “Yes” (employees who left).
- TP: The model correctly predicted 244 instances as “No” (employees who did not leave).
- FP: The model incorrectly predicted 2 instances as “Yes” when it was actually “No”.
- FN: The model incorrectly predicted 39 instances as “No” when they were actually “Yes”.

The Random Forest model has a high number of True Positives (244), correctly identifying employees who have not left the company. However, it has a relatively low number of True Negatives (9), indicating that it correctly identifies only a small proportion of employees who have left. The model has a very low number of False Positives (2), which means it rarely misclassifies employees who have not left as having left. On the other hand, the model has a higher number of False Negatives (39), incorrectly classifying employees who have left as still being with the company. This suggests that the model may have difficulty capturing all the instances of employee turnover.

Table 3. Confusion Matrix for Random Forest

	Predicted “No”	Predicted “Yes”
Actual “No”	244	2
Actual “Yes”	39	9

b. Logistic Regression:

- TN: The model correctly predicted 27 instances as “Yes” (employees who left).
- TP: The model correctly predicted 237 instances as “No” (employees who did not leave).
- FP: The model incorrectly predicted 21 instances as “Yes” when they were actually “No”.
- FN: The model incorrectly predicted 9 instances as “No” when they were actually “Yes”.

The Logistic Regression model has a good balance between True Positives (237) and True Negatives (27), indicating decent overall accuracy. It has a relatively low number of False Positives (21) and False Negatives (9). This model seems to have a balanced performance in identifying both positive and negative instances.

Table 4. Confusion Matrix for Logistic Regression

	Predicted “No”	Predicted “Yes”
Actual “No”	237	21
Actual “Yes”	9	27

c. Artificial Neural Networks (ANN):

- TN: The model correctly predicted 26 instances as “Yes” (employees who left).
- TP: The model correctly predicted 235 instances as “No” (employees who did not leave).
- FP: The model incorrectly predicted 22 instances as “Yes” when they were actually “No”.
- FN: The model incorrectly predicted 11 instances as “No” when they were actually “Yes”.

The Artificial Neural Networks model shows a high number of True Positives (235), accurately identifying employees who have not left. It has a relatively low number of False Positives (22), minimizing the misclassification of employees who have left as still being with the company. The model has a moderate number of True Negatives (26) and False Negatives (11), demonstrating a reasonable ability to identify employees who have left.

Table 5. Confusion Matrix for ANN

	Predicted “No”	Predicted “Yes”
Actual “No”	235	22
Actual “Yes”	11	26

d. Support Vector Machines (SVM):

- TN: The model correctly predicted 4 instances as “Yes” (employees who left).
- TP: The model correctly predicted 246 instances as “No” (employees who did not leave).
- FP: The model incorrectly predicted 44 instances as “Yes” when they were actually “No”.
- FN: The model incorrectly predicted 0 instance as “No” when they were actually “Yes”.

The SVM model has a high number of True Positives (246), accurately identifying employees who have not left. However, it also has a high number of False Positives (44), suggesting that it often misclassifies employees who have left as still being with the company. The model has a low number of True Negatives (4) and False Negatives (8), indicating poor performance in correctly identifying employees who have left.

Table 6. Confusion Matrix for SVM

	Predicted “No”	Predicted “Yes”
Actual “No”	246	44
Actual “Yes”	0	4

e. XGBoost:

- TN: The model correctly predicted 12 instances as “Yes” (employees who left).
- TP: The model correctly predicted 244 instances as “No” (employees who did not leave).
- FP: The model incorrectly predicted 36 instance as “Yes” when it was actually “No”.
- FN: The model incorrectly predicted 2 instances as “No” when they were actually “Yes”.

The XGBOOST model has a high number of True Positives (244), correctly identifying employees who have not left. However, it also has a relatively high number of False Positives (36), indicating a tendency to misclassify employees who have left as still being with the company. The model has a low number of True Negatives (12) and False Negatives (2), suggesting difficulty in accurately identifying employees who have left.

Table 7. Confusion Matrix for XGBoost

	Predicted "No"	Predicted "Yes"
Actual "No"	244	36
Actual "Yes"	2	12

f. Decision Tree:

- TN: The model correctly predicted 16 instances as "Yes" (employees who left).
- TP: The model correctly predicted 237 instances as "No" (employees who did not leave).
- FP: The model incorrectly predicted 32 instance as "Yes" when it was actually "No".
- FN: The model incorrectly predicted 9 instances as "No" when they were actually "Yes".

The Decision Tree model has a relatively balanced performance. It has a good number of True Positives (237), correctly identifying employees who have not left. The number of False Positives (32) is moderate, showing some misclassification of employees who have left as still being with the company. The True Negatives (10) and False Negatives (9) are relatively balanced, indicating a fair ability to identify employees who have left.

Table 8. Confusion Matrix for Decision Tree

	Predicted "No"	Predicted "Yes"
Actual "No"	237	32
Actual "Yes"	9	16

From the confusion matrices, we can see that the Artificial Neural Networks and Logistic Regression models exhibit a more balanced performance in correctly identifying both employees who have not left and those who have left. The Random Forest model performs well in identifying employees who have not left but may struggle to capture all instances of employee turnover. The XGBoost and Decision Tree models show a tendency to misclassify employees who have left as still being with the company, while the SVM model exhibits a strong bias towards predicting employees as staying with the company.

These confusion matrices provide insights into the models' performance and can help in identifying areas for improvement, such as addressing class imbalance or tuning the models to better identify employee turnovers and retetntions.

4.2 ML Model Testing Results

The results from the ML model testing within the framework of the accuracy, sensitivity, specificity, precision, F1 Score and ROC-AUC metrics are presented in the Table 10.

Based on the results, the Logistic Regression model outperformed the other models in terms of accuracy (89.80%), sensitivity (96.34%), and F1 Score (0.5614). It also achieved a high ROC-AUC value of 0.902, indicating its strong overall performance in distinguishing between the positive and negative classes.

The Artificial Neural Networks model also demonstrated good performance, with an accuracy of 88.78%, sensitivity of 95.53%, and the highest F1 Score among all models (0.6364). However, its ROC-AUC value (0.784) was lower compared to the Logistic Regression and Random Forest models.

The Random Forest model achieved an accuracy of 86.05% and a high ROC-AUC value of 0.900. It exhibited balanced performance in terms of sensitivity (86.22%) and specificity (81.82%). However, its F1

Table 9. Results for all ML models

Model	**Accuracy**	**Sensitivity**	**Specificity**	**Precision**	**F1 Score**	**ROC-AUC**
Random Forest	0.8605	0.8622	0.8182	0.9919	0.3067	0.900
Logistic Regression	0.898	0.9634	0.5625	0.9186	0.5614	0.902
Artificial Neural Networks	0.8878	0.9553	0.5417	0.9144	0.6364	0.784
SVM	0.8503	1	0.08333	0.8483	0.1538	0.524
XGBoost	0.8707	0.9919	0.25	0.8714	0.3871	0.851
Decision Tree	0.8605	0.9634	0.3333	0.8810	0.439	0.684

Score (0.3067) was relatively lower compared to the Logistic Regression and Artificial Neural Networks models.

The XGBoost model showed an accuracy of 87.07% and a high sensitivity of 99.19%, indicating its effectiveness in correctly identifying positive instances. However, its specificity (25%) and F1 Score (0.3871) were lower compared to the other models.

The Decision Tree model achieved an accuracy of 86.05%, similar to the Random Forest model. It demonstrated high sensitivity (96.34%) but relatively lower specificity (33.33%) and F1 Score (0.439).

The SVM model exhibited the lowest accuracy (85.03%) among all models. While it achieved perfect sensitivity (100%), its specificity (8.33%) and F1 Score (0.1538) were the lowest, indicating a high rate of false positives.

The ROC-AUC values provide an aggregate measure of each model's performance, considering both its ability to identify positive instances (employee retention) and negative instances (employee turnover). The Logistic Regression and Random Forest models achieved the highest ROC-AUC values (0.902 and 0.900, respectively), indicating their superior overall performance compared to the other models.

It is important to note that the presence of class imbalance in the dataset can influence the models' performance, particularly in terms of sensitivity and F1-score for the minority class (employee turnover). Addressing class imbalance through techniques such as oversampling, undersampling, or using class weights can help improve the models' ability to correctly identify instances of employee turnover.

Also, one limitation of this study is the reliance on a single dataset. While the "IBM HR Analytics Employee Attrition & Performance" dataset provides a diverse set of employee records, the results' generalizability to other organizations or industries may be limited. Future research could validate the findings using datasets from different contexts or conduct multi-organizational studies to assess the models' performance across various settings.

From a practical standpoint, the findings of this study have several implications for businesses aiming to leverage ML techniques for employee turnover prediction as the study presents in the following section.

4.3 ML Tools Suitable for Employee Turnover Prediction

Based on the performance metrics evaluated in this study, the following machine learning tools are considered suitable for employee turnover prediction:

The Logistic Regression model demonstrated the highest accuracy, sensitivity, and F1 Score, along with a high ROC-AUC value. It is a simple and interpretable model that can provide insights into the factors contributing to employee turnover. Logistic Regression is particularly suitable when the relationship between the predictors and the target variable is linear.

The Artificial Neural Networks model achieved the second-highest accuracy and the highest F1 Score. It is capable of capturing complex non-linear relationships between the predictors and the target variable. Artificial Neural Networks can be effective when dealing with large datasets and when the underlying relationships are not well understood.

The Random Forest model exhibited balanced performance in terms of sensitivity and specificity, along with a high ROC-AUC value. It is an ensemble learning method that combines multiple decision trees, making it robust to outliers and noise. Random Forest can handle both categorical and numerical predictors and can provide feature importance rankings.

The XGBoost model demonstrated high sensitivity and a relatively high ROC-AUC value. It is an optimized implementation of gradient boosting that can handle complex interactions among predictors. XGBoost is known for its excellent predictive performance and its ability to handle missing values.

The SVM and Decision Tree models had lower overall performance compared to the above models, but they may still be considered in certain scenarios. SVMs can be effective when dealing with high-dimensional data, while Decision Trees offer interpretability and can handle both categorical and numerical predictors.

When selecting the most suitable ML tool for predicting employee turnover, it is essential to consider factors such as the size and complexity of the dataset, the interpretability requirements, the presence of non-linear relationships, and the computational resources available. It is also recommended to experiment with multiple tools and compare their performance using appropriate evaluation metrics to determine the best approach for the specific dataset and problem at hand.

The results and discussion section has provided valuable insights into the performance of various ML models for predicting employee turnover. In the following conclusion, the authors will summarize the key findings, discuss the implications of our study, and outline potential avenues for future research in this domain.

6. Conclusion

This study aimed to leverage ML techniques within the framework of HRA to predict employee turnover effectively. By evaluating and comparing the performance of Random Forest, Logistic Regression, Artificial Neural Networks, Support Vector Machines, XGBoost and Decision Tree models on the "IBM HR Analytics Employee Attrition & Performance" dataset, the study contributes to the existing body of knowledge in HRA and employee turnover prediction.

The findings suggest that the Logistic Regression model can be an effective tool in human resources analytics for turnover prediction. However, the choice of model should be based on the specific use case, considering the strengths and weaknesses of each model. Organizations should evaluate their requirements and prioritize the relevant performance metrics when selecting a model for implementation.

The findings of this study have practical implications for businesses seeking to implement data-driven strategies for workforce management. By leveraging ML techniques, organizations can proactively identify employees at risk of turnover and take appropriate measures to retain valuable talent. However, Organizations implementing ML models for employee turnover prediction should also consider the ethical implications and potential biases associated with these approaches. Ensuring fairness, transparency, and privacy in the use of employee data is crucial to maintain trust and comply with legal and ethical standards.

Future research directions include exploring alternative ML models, investigating the impact of feature selection techniques, and addressing class imbalance to further improve the predictive performance of the models. Additionally, validating the findings using datasets from different contexts or conducting multi-organizational studies can enhance the generalizability of the results.

In conclusion, this study demonstrates the potential of ML techniques within the HRA framework for predicting employee turnover. By continuously refining and improving these models, businesses can make data-driven decisions to optimize their workforce planning, reduce turnover costs, and enhance overall employee satisfaction and retention.

References

Aarons, G., Sawitzky, A., 2006. Organizational climate partially mediates the effect of culture on work attitudes and staff turnover in mental health services. Administration and Policy in Mental Health and Mental Health Services Research, 33(3), 289-301. https://doi.org/10.1007/s10488-006-0039-1

Akter, S., Wamba, S. F., Gunasekaran, A., Dubey, R., Childe, S. J., 2016. How to improve firm performance using big data analytics capability and business strategy alignment? International Journal of Production Economics, 182, 113–131. https://doi.org/10.1016/j.ijpe.2016.08.018

Alan, A., 2020. Makine Öğrenmesi Sınıflandırma Yöntemlerinde Performans Metrikleri ile Test Tekniklerinin Farklı Veri Setleri Üzerinde Değerlendirilmesi (Yüksek Lisans Tezi). Fırat Üniversitesi, Fen Bilimleri Enstitüsü, s.19

Alsaadi, E., Khlebus, S., Alabaichi, A., 2022. Identification of Human Resoıurce Analytics using MLalgorithms. Telkomnika (Telecommunication Computing Electronics and Control), 20(5), 1004. https://doi.org/10.12928/telkomnika.v20i5.21818

Ashworth, M., 2006. Preserving knowledge legacies: workforce aging, turnover and human resource issues in the us electric power industry. The International Journal of Human Resource Management, 17(9), 1659-1688. https://doi.org/10.1080/09585190600878600

Avrahami, D., Pessach, D., Singer, G., Ben-Gal, H. C., 2022. A human resources analytics and machine-learning examination of turnover: implications for theory and practice. International Journal of Manpower, 43(6), 1405-1424. https://doi.org/10.1108/ijm-12-2020-0548

Bahadır, M. B., Bayrak, A. T., Yücetürk, G., Ergun, P., 2021. A Comparative Study for Employee Churn, Prediction, Researchgate, 1-4.

Balcıoğlu, Y. S., Artar, M., 2022. Çalışanların İşten Ayrılma Olasılığının Makine Öğrenmesi İle Tahmini: K-En Yakın Komşu Algoritması İle. Güncel İşletme, Yönetim ve Muhasebe Çalışmaları, 29-35. https://www.researchgate.net/publication/359362785

Bardoel, A., Russell, G., Advocat, J., Mayson, S., Kay, M., 2020. Turnover among australian general practitioners: a longitudinal gender analysis. Human Resources for Health, 18(1). https://doi.org/10.1186/s12960-020-00525-4

Barın, H. D., 2022. Employee Turnover Probability Prediction, A thesis submitted to the Graduate School of Engineering and Science of Bilkent University for the degree of Master of Science in Industrial Engineering, 1-75.

Belbin, C., Erwee, R., Wiesner, R., 2012. Employee perceptions of workforce retention strategies in a health system. Journal of Management & Organization, 18(5), 742-760. https://doi.org/10.5172/jmo.2012.18.5.742

Bogaert, K., Leider, J., Castrucci, B., Sellers, K., Whang, C., 2019. Considering leaving, but deciding to stay: a longitudinal analysis of intent to leave in public health. Journal of Public Health Management and Practice, 25(2), S78-S86. https://doi.org/10.1097/phh.0000000000000928

Breiman, L., 2001. Rastgele Ormans. Machine learning, 45(1), 5-32.

Catani F, Lagomarsino D, Segoni S, Tofani V., 2013. Landslide susceptibility estimation by Rastgele Ormans technique: sensitivity and scaling issues. Nat Hazards Earth Syst Sci, 13:2815–2831, doi:10.5194/nhess-13-2815-2013.

Chapman, G., Nasirov, S., Özbilgin, M., 2022. Workforce diversity, diversity charters and collective turnover: long-term commitment pays. British Journal of Management, 34(3), 1340-1359. https://doi.org/10.1111/1467-8551.12644

Chaudhary, M., 2022. Rationale of employee turnover: an analysis of banking sector in nepal. International Research Journal of MMC, 3(2), 18-25. https://doi.org/10.3126/irjmmc.v3i2.46291

Chisholm, M., Russell, D., Humphreys, J., 2011. Measuring rural allied health workforce turnover and retention: what are the patterns, determinants and costs?. Australian Journal of Rural Health, 19(2), 81-88. https://doi.org/10.1111/j.1440-1584.2011.01188.x

Choi, J., Ko, I., Kim, J., Jeon, Y., Han, S., 2021. MLframework for multi-level classification of company revenue. Ieee Access, 9, 96739-96750. https://doi.org/10.1109/access.2021.3088874

Demir, K., Çalık, E., 2021. İnsan Kaynakları Analitiği: Modelleme ve Örnek Uygulamalarla. 2. Baskı, Nobel Bilimsel Yayıncılık.

Emmanuel-Okereke, I. L., Anigbogu, S. O., 2022. Predicting the Perceived Employee Tendency of Leaving an

Organization Using SVM and Naive Bayes Techniques. Open Access, 1-15.

Erkal, H., Keçecioğlu, T., Yılmaz, M. K., 2014. Gelecek 10 Yıl İçerisinde İnsan Kaynaklarının Yüzleşeceği Zorluklar. EUL Journal of Social Sciences, V(II), LAÜ Sosyal Bilimler Dergisi, Aralık, 32-63.

Gallup., 2023. State of the Global Workplace Report - Gallup. Gallup.com. Retrieved February 21, 2024, from https://www.gallup.com/workplace/349484/state-of-the-global-workplace.aspx#ite-506924

Gao, X., Wen, J., Zhang, C., 2019. An improved random forest algorithm for predicting employee turnover. Mathematical Problems in Engineering, 2019, 1-12. https://doi.org/10.1155/2019/4140707

Hatch-Maillette, M., Harwick, R., Baer, J., Masters, T., Cloud, K., Peavy, M., Wells, E., 2019. Counselor turnover in substance use disorder treatment research: observations from one multisite trial. Substance Abuse, 40(2), 214-220. https://doi.org/10.1080/08897077.2019.1572051

Healy, K., Oltedal, S., 2010. An institutional comparison of child protection systems in australia and norway focused on workforce retention. Journal of Social Policy, 39(2), 255-274. https://doi.org/10.1017/s004727940999047x

https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset?resource=download

Jain, P. K., Jain, M., Pamula, R., 2020. Explaining and Predicting Employees' Attrition: A MLApproach. Research Article, 1-11.

Judrups, J., Cinks, R., Birzniece, I., Andersone, I., 2021. MLbased solution for predicting voluntary employee turnover in organization.. https://doi.org/10.22616/erdev.2021.20.tf296

Karcı, Z., 2017. Lojistik Regresyon Modeli ile Elde Edilen Tahminlerin ROC Eğrisi Yardımıyla Değerlendirilmesi: Türkiye'de Hanehalkı Yoksulluğu Üzerine Bir Araştırma (Yüksek Lisans Tezi). T.C. Süleyman Demirel Üniversitesi Sosyal Bilimler Enstitüsü, Ekonometri Anabilim Dalı, Isparta, 46.

Kışaoğlu, Z. Ö., 2014. Employee Turnover Prediction Using MLBased Methods, A thesis submitted to the Graduate School of Natural and Applied Sciences of Middle East Technical University.

Kropp, B., McRae, E. R., 2022. 11 Trends that Will Shape Work in 2022 and Beyond, 11 Trends that Will Shape Work in 2022 and Beyond (hbr.org)

Kutlugün, M. A., Çakır, M. Y., Kiani, F., 2017. Yapay Sinir Ağları ve K-En Yakın Komşu Algoritmalarının Birlikte Çalışma Tekniği (Ensemble) ile Metin Türü Tanıma, 2 https://www.researchgate.net/publication/323990877.

Liao, C., 2023. Employee turnover prediction using MLmodels.. https://doi.org/10.1117/12.2672733

Liu, H. and Liu, Y., 2021. Visualization research and analysis of turnover intention. E3s Web of Conferences, 253, 02018. https://doi.org/10.1051/e3sconf/202125302018

Maharjan, R., 2021. Employee Churn Prediction using Logistic Regression and Support Vector Machine, San Jose State University, Master's Projects. DOI: https://doi.org/10.31979/etd.3t5h-excq.

Maisuradze, M., 2017. Predictive analysis on the example of employee turnover (Master's thesis). Tallinn University of Technology, Faculty of Information Technology, Department of Computer Systems, 3-76.

Mayson, S., Bardoel, A., 2021. Sustaining a career in general practice: embodied work, inequality regimes, and turnover intentions of women working in general practice. Gender Work and Organization, 28(3), 1133-1151. https://doi.org/10.1111/gwao.12659

McCarthy, A., Moonesinghe, R., Dean, H., 2020. Association of employee engagement factors and turnover intention among the 2015 u.s. federal government workforce. Sage Open, 10(2), 215824402093184. https://doi.org/10.1177/2158244020931847

Moturi, D. G., Wekesa, S., Juma, D., 2023. Influence of self efficacy on employee acceptance levels and use of human resource analytics in microfinance institutions in kenya. International Journal of Business Management, Entrepreneurship and Innovation, 5(1), 31-50. https://doi.org/10.35942/jbmed.v5i1.304

Onnis, L., 2017. Human Resourse Management policy choices, management practices and health workforce sustainability: remote australian perspectives. Asia Pacific Journal of Human Resources, 57(1), 3-23. https://doi.org/10.1111/1744-7941.12159

Pavansubhash, 2016. IBM HR Analytics Employee Attrition & Performance

Peryön., 2018. Çalişan Devir Orani Araştirmasi Sonuç Raporu. In https://www.peryon.org.tr/upload/files/PERYO%CC%88N_C%CC%A7al%C4%B1s%CC%A7an_Devir_Oran%C4%B1_Sonuc%CC%A7_Raporu_2017-2018.pdf.

Poku, C., Alem, J., Poku, R., Osei, S., Amoah, E., Ofei, A., 2022. Quality of work-life and turnover intentions among the ghanaian nursing workforce: a multicentre study. Plos One, 17(9), e0272597. https://doi.org/10.1371/journal.pone.0272597

Putri, M. and Rachmawati, R., 2022. Psychological contract, employee engagement, and perceived organizational support influence on employee turnover intention in pharmaceutical industry.. https://doi.org/10.4108/eai.27-7-2021.2316894

Randstad., 2022. Randstand Trends 2022 Report. In https://www.randstad.gr/. Retrieved February 21, 2024, from https://www.randstad.gr/s3fs-media/gr/public/2022-07/hr-trends-2022-salary-report-eng.pdf

Randstad., 2023. Randstand Trends 2023 Report. In https://www.randstad.com.tr/. Retrieved February 21, 2024, from https://www.randstad.com.tr/s3fs-media/tr/public/2023-04/TR_Turkey%20HR%20Trends%202023_0.pdf

Ribes, E., Touahri, K., Perthame, B., 2017. Employee turnover prediction and retention policies design: a case study, 1-10.

Roche, A., McEntee, A., Kostadinov, V., Hodge, S., Chapman, J., 2021. Older workers in the alcohol and other drug sector: predictors of workforce retention. Australasian Journal on Ageing, 40(4), 381-389. https://doi.org/10.1111/ajag.12917

Rokach, L., Maimon, O., 2005. Decision Trees. In O. Maimon & L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook (pp. 165-192). Springer US. https://doi.org/10.1007/0-387-25465-X_9

Russell, D., Zhao, Y., Guthridge, S., Ramjan, M., Jones, M., Humphreys, J. Wakerman, J., 2017. Patterns of resident health workforce turnover and retention in remote communities of the northern territory of australia, 2013–2015. Human Resources for Health, 15(1). https://doi.org/10.1186/s12960-017-0229-9

Scanlan, J., Meredith, P., Poulsen, A., 2013. Enhancing retention of occupational therapists working in mental health: relationships between wellbeing at work and turnover intention. Australian Occupational Therapy Journal, 60(6), 395-403. https://doi.org/10.1111/1440-1630.12074

Schlechter, A., Syce, C., Bussin, M., 2016. Predicting voluntary turnover in employees using demographic characteristics: a south african case study. Acta Commercii, 16(1). https://doi.org/10.4102/ac.v16i1.274

Shanthakumara, A. H., Divya, J., Harshitha, H. T., Pallavi, L. V., Spoorthy, B. C. S., 2022. Prediction of Employee Turnover using Machine Learning. Grenze Scientific Society, 1-13.

Shrivastava, S., Nagdev, K., Rajesh, A., 2017. Redefining HR using people analytics: the case of Google. Human Resourse Management International Digest, 1-4.

Stachová, K., Baroková, A., Stacho, Z., 2021. Optimization of Employee Turnover through Predictive Analysis, Institut of Management, University of Ss. Cyril and Methodius in Trnava, Slovakia. Faculty of Management, Comenius University, Bratislava, Slovakia.

State of the Global Workplace Report - Gallup., 2024. Gallup.com. https://www.gallup.com/workplace/349484/state-of-the-global-workplace.aspx#ite-506924

Uzak, B., 2022. Telekomünikasyon Sektöründe Çalışan Kaybı Tahmini İçin Makine Öğrenmesi Modeli Seçimi (Yüksek Lisans Tezi). T.C. Bursa Uludağ Üniversitesi Fen Bilimleri Enstitüsü, 4-5.

Van Vulpen, E., 2023. What is HR Analytics? All You Need to Know to Get Started. AIHR. https://www.aihr.com/blog/what-is-hr-analytics/

Wijaya, D., Ds, J., Barus, S., Pasaribu, B., Sirbu, L., Dharma, A., 2021. Uplift modeling vs conventional predictive model: a reliable MLmodel to solve employee turnover. International Journal of Artificial Intelligence Research, 5(1). https://doi.org/10.29099/ijair.v4i2.169

Woltmann, E., Whitley, R., McHugo, G., Brunette, M., Torrey, W., Daras, L., Drake, R., 2008. The role of staff turnover in the implementation of evidence-based practices in mental health care. Psychiatric Services, 59(7), 732-737. https://doi.org/10.1176/ps.2008.59.7.732

Yavuz, H. V., 2016, Sanayi ve Hhizmet sektöründe işgücü devir oranlarinin yüksek olmasinin nedenleri ve çözüm önerileri: Denizli örneği (Yüksek Lisans Tezi). Pamukkale Üniversitesi Sosyal Bilimler Enstitüsü. Yüksek Lisans Tezi, Çalışma Ekonomisi ve Endüstri İlişkileri Anabilim Dalı, DENİZLİ, 5-14.

Ye, J., Pu, B., Guan, Z., 2019. Entrepreneurial leadership and turnover intention in startups: mediating roles of employees' job embeddedness, job satisfaction and affective commitment. Sustainability, 11(4), 1101. https://doi.org/10.3390/su11041101

Yedida, R., Reddy, R., Vahi, R., J, R., Abhilash, Kulkarni, D., 2018. Employee Attrition Prediction, https://www.academia.edu/73094870/Employee_Attrition_Prediction, 1-3.

Zhu, Q., Shang, J., Cai, X., Jiang, L., Liu, F., Qiang, B., 2019. CoxRF: Employee Turnover Prediction based on Survival Analysis. In Proceedings of the 2019 IEEE, 1123-1130.

Araştırma Makalesi
Zeki Sistemler Teori ve Uygulamaları Dergisi 7(2) (2024) 159-173
DOI: 10.38016/jista.1509968

Fenomen-Hedef Kitle Eşleştirmesinin Otomatikleştirilmesi: Sosyal Medya Gönderilerinin Sınıflandırılması ile Reklama Yönelik Hedef Kitle Analizi

Mehmet Varan[1], Aslı Yatkınoğlu[2*], Amine Gonca Toprak[3], Fatih Soygazi[4], Bora Mocan[5]

[1, 2, 3, 5] AdresGezgini A.Ş., İzmir, Türkiye

[4] Bilgisayar Mühendisliği Bölümü, Adnan Menderes Üniversitesi, Aydın, Türkiye

mehmetvaran@adresgezgini.com, aslicankut@adresgezgini.com, goncatoprak@outlook.com, fatih.soygazi@adu.edu.tr, boramocan@adresgezgini.com

Öz

İnternet kullanımının son yıllarda yaygınlaşması, bireylerin ve toplumların iletişimden alışveriş alışkanlıklarına kadar neredeyse her alanda davranışlarının evrilerek büyük değişikliklerin ortaya çıkmasına sebep olmuştur. Böylece geleneksel iletişim yöntemleri de dönüşüme uğramıştır. Bu gelişmeler sonucunda, günümüzde en yaygın iletişim aracı olarak kabul edilen sosyal medya kavramı doğmuştur. Yeni bir iletişim şekli olan sosyal medya, kurum ve kuruluşların hedef kitleleri ile yer ve zaman kısıtı olmaksızın doğrudan iletişim kurabilmelerini mümkün kılarak reklam verenler için ürünlerini tanıtabilecekleri oldukça etkili bir kanal haline gelmiştir. Sosyal medyada ürün pazarlamak "fenomen" olarak adlandırılan kişiler sayesinde gerçekleşmektedir ve her fenomenin hitap ettiği bir hedef kitle bulunmaktadır. Bu bağlamda, fenomenlerin hitap ettiği hedef kitle ile reklamı yapılacak ürünün hedef kitlesinin doğru bir şekilde eşleşmesi, sosyal medya üzerinden yapılan ürün pazarlamasında kritik bir rol oynamaktadır. Bu çalışmada en doğru fenomen-ürün hedef kitle eşleşmesini gerçekleştirebilmek adına, Instagram fenomenlerinin paylaşmış olduğu gönderileri analiz ederek fenomenin hedef kitlesini kategorize eden bir metin sınıflandırma modeli geliştirilmiştir. Bu amaç doğrultusunda veri gizliliğini ihlal etmemek adına Instagram profili herkese açık olan 1.005 farklı fenomenin üçüncü taraf bir yazılım ile gönderileri elde edilerek bu gönderilerdeki açıklamalar BERTopic mimarisi ile kümelenmiştir. Oluşturulan kümelerin temsilleri ve içeriği incelenerek temsil ettiği kategoriye göre etiketlenmiştir. Etiketlenen veriler ile BERTurk sınıflandırma modeli geliştirilmiştir. Sınıflandırma model performans değerlendirilmesi sonucunda ölçülerek 0,92 doğruluk ve 0,91 F1 skor değeri elde edilmiştir. Elde edilen sonuçlar doğrultusunda yüksek sınıflandırma doğruluğu ile fenomen gönderilerini otomatik olarak kategorize edebilen bir sistem geliştirilmiş ve fenomen-ürün hedef kitle eşleştirilmesinde başarıyla kullanılmıştır.

Anahtar Kelimeler: Metin sınıflandırma, Kümeleme Analizi, BERTopic, BERTurk, Instagram

Automating Influencer-Target Audience Matching: Target Audience Analysis for Advertising through Classification of Social Media Posts

Abstract

The widespread adoption of the internet has led to significant transformations in individual and societal behaviors, influencing everything from communication to shopping habits. As a result, traditional communication methods have evolved, giving rise to social media as a dominant medium today. Social media enables organizations to engage directly with target audiences without geographical or temporal constraints, making it an effective platform for advertisers. Social media marketing is often facilitated by "influencers," individuals who have built their own audience. Accurate matching between the influencer's target audience and the advertised product's audience is essential for effective social media marketing. This study aims to develop a text classification model that categorizes the target audiences of Instagram influencers by analyzing their posts, in order to achieve the most accurate influencer-product target audience matching. To avoid violating data privacy, posts from 1.005 distinct influencers with publicly accessible Instagram profiles were collected using a third-part software, and the descriptions in these posts were clustered using the BERTopic architecture for topic modeling. A BERTurk classification model was developed using the labeled data. The representations and content of the resulting clusters were analyzed and labeled according to the categories they represented. These labeled records were then used for classification purposes. The performance of the classification model was evaluated, achieving an

* Sorumlu yazar
E-posta adresi: aslicankut@adresgezgini.com

Gönderme : 4 Temmuz 2024
Revizyon : 12 Ağustos 2024
Kabul : 7 Eylül 2024

accuracy of 0,92 and an F1 score of 0,91. The results demonstrate the development of a system that can automatically categorize influencer posts with high classification accuracy and has been successfully applied for influencer-product target audience matching.

Keywords: Text Classification, Clustering Analysis, BERTopic, BERTurk, Instagram

1. Giriş (Introduction)

Gün geçtikçe teknolojinin gelişmesi ve bireylerin günlük yaşamlarında internetin yaygınlaşmasıyla birlikte arkadaşlıklar, gündem takibi, boş zaman değerlendirme, alışveriş gibi günlük aktiviteler internet ortamına da taşınmıştır (Yıldırım ve Yıldırım, 2022). Özellikle mobil cihaz (akıllı telefon, tablet vb.) kullanımının artmasıyla internet yer ve zaman fark etmeksizin ulaşılabilir bir hale gelerek bireylerin internette daha fazla vakit geçirmesine sebep olmuştur. Teknolojinin ve internetin günlük yaşamda bu denli büyük bir role sahip olması, insanlık tarihi boyunca sürekli değişerek evrilen iletişim yöntem ve kanallarını da etkileyerek sosyal medyayı doğurmuştur (Şahinkayası ve Şahinkayası, 2017).

Sosyal medya bireylerin birbirleriyle video, mesaj veya fotoğraf içerikleriyle paylaşımda bulunmalarına ve iletişim kurmalarına olanak sağlayan çeşitli çevrimiçi platformlar olarak tanımlanabilir (Carr ve Hayes, 2015). Günümüzde sosyal medya bireylerin daha fazla iletişime geçmesine olanak sağlayarak vazgeçilmez bir iletişim aracı haline gelmiştir. Sosyal medya platformlarının yaygınlaşmasıyla birlikte farklı yaş gruplarından oluşan sosyal medya kullanıcıları, çeşitli ilgi alanları doğrultusunda hedef kitleleri oluşturmaktadır.

Sosyal medyanın en çok etkilediği alanlardan bir tanesi ürün pazarlama faaliyetleridir (Terkan, 2014). Geleneksel ürün pazarlama yöntemlerine kıyasla sosyal medya platformları, işletme büyüklüğü ve sektör fark etmeksizin tüm işletmelere her farklı yaş grubu ve sosyo-ekonomik gruptan tüketicilerle yani hedef kitlelerle iletişim kurabilme olanağı sağlamaktadır (Arslan, 2017). Instagram, Facebook, Twitter, Youtube, Tiktok, Linkedin gibi farklı sosyal medya ağları üzerinden oluşturulan çeşitli içeriklerle işletmeler, sosyal medyanın sağladığı çift taraflı ve etkileşimli iletişim sayesinde hedef kitlelere daha etkili ve daha az maliyetli bir şekilde doğrudan ulaşabilmektedir.

Ürün pazarlamanın temel araçlarından biri olan reklam, bir ürün veya hizmetin medya kanalları aracılığıyla kitlelere tanıtılması olarak tanımlanmaktadır (Bagwell, 2007). Farklı kanallar üzerinde belirli bir ücret karşılığında yapılan reklamların temel amacı, tüketicilerin ilgisini reklamı yapılan ürün veya hizmete yönlendirerek ilgili ürün veya hizmet satışının gerçekleştirilmesidir. Gazete, televizyon, radyo, dergi gibi farklı iletişim kanalları aracılığıyla yayınlanan reklamların günümüzde internetin etkisi ile sosyal medyada yaygınlaşmasıyla, sosyal medya araçları önemli bir ürün pazarlama aracı haline evrilmiştir (Özdemir vd., 2014).

Sosyal medya üzerinden ürün tanıtımları dijital reklamlar ve sosyal medya ağlarında fazla sayıda takipçisi olan hesaplar olarak adlandırılan fenomenler üzerinden gerçekleşmektedir.

Sosyal medya fenomenleri, oluşturdukları içerikler ve yaptıkları paylaşımlarla geniş kitlelere ulaşarak bireylerin düşünce, tutum ve davranışlarını etkileyebilmektedir. Sosyal medya fenomenlerinin aynı zamanda tüketicilerin satın alma kararları üzerinde de önemli bir etkisinin olması, ürün pazarlama literatürüne yeni bir kavram kazandırmıştır. Literatürde sosyal medya platformlarında fenomenler aracılığıyla gerçekleştirilen ürün pazarlama faaliyetleri şeklinde tanımlanan bu yeni kavram, fenomen pazarlaması olarak adlandırılmaktadır (Leung vd., 2022).

Fenomen pazarlaması, fenomenin herhangi bir sosyal medya platformu aracılığıyla bir ürün veya hizmete dair sunduğu, tüketicinin satın alma motivasyonunu etkileyen pazarlama aktiviteleri üzerinden gerçekleşmektedir. Tüketiciler bir ürün veya hizmet satın alırken gerçek tüketici deneyimlerine çok önem verdiğinden fenomen pazarlaması, işletmelerin hedef kitlelere ulaşabilmek için tercih ettiği en yaygın ürün pazarlama yöntemlerinden biri haline gelmiştir (Çopuroğlu, 2022). Bir diğer deyişle, fenomenler, sosyal medya platformları üzerinde işletmelerin reklam yüzü olarak içerik oluşturduğu alanda (ör. seyahat, yemek, moda vb.) hedef kitlesinin satın alma motivasyonunu olumlu bir şekilde etkilemek için içerik üreterek iş birliği yaptığı işletmenin ürün veya hizmetinin tüketicilere ulaşmasını sağlamaktadır. İşletmelerin, tüketicilerin satın alma niyetini olumlu yönde etkilemek istedikleri takdirde fenomenler ile iş birliği yaparak kazanç sağlayabilecekleri görülmüştür. Yapılan araştırmalarda fenomenlerin reklamlarına yönelik pozitif bir tutum olduğu, satışları arttırmaya yönelik olumlu etkileri olduğu ortaya çıkmıştır (Karataş ve Eti, 2022).

Fenomen pazarlamasının başarısı için en önemli faktörlerden biri, iş birliği yapacak olan fenomen ve işletmenin hedef kitlesinin örtüşmesidir (Öztek vd., 2021). Örneğin, mobilya üreten bir işletmenin hedef kitlesi kozmetik alanında ürün pazarlamaya daha uygun olan bir fenomen ile iş birliği yapması, ürün pazarlamada istenen getiriyi sağlamayacaktır. İşletme veya marka ile fenomen arasında gerçekleştirilecek iş birliğinde, fenomenin sosyal medya platformundaki paylaşımlarının içerik analizi yoluyla incelenmesi sonucunda, fenomenin hitap ettiği hedef kitle tespit edilebilir. Literatürde doğal dil işleme, yapay zeka ile sınıflandırma gibi güncel çalışma alanları sayesinde sosyal medyadaki fenomenlerin gönderileri gözetilerek otomatik bir şekilde gerçekleştirilebilmektedir (Kim vd., 2020).

Bu çalışmada Instagram fenomenlerinin sosyal medya hesap gönderilerindeki açıklamaların (caption) doğal dil işleme yöntemleri ile analiz edilerek hitap ettiği kitlenin tespiti amaçlanmıştır. Bu kapsamda öncelikle Apify (Apify, 2022) web veri çıkarma (web scraping) platformu ile Instagram'daki halka açık olan hesap gönderilerinin elde edilmesiyle bir veri seti oluşturulmuştur. Açıklama içermeyen gönderilerin ayrılması gibi veri ön işleme adımları ile veri seti, dil modellerinin eğitimine hazır hale getirilmiştir. Elde edilen veri setindeki açıklamalar, topik modelleme algoritması kullanılarak kümelenmiştir. BERTopic modelinin ürettiği kümeler, veri etiketleme ekibi tarafından titizlikle incelenmiş ve her bir kümenin içeriğine uygun etiketler önceden belirlenmiş olan 18 kategoriye göre etiketlenmiştir. Bu süreçte, kümelerin temsil ettiği temalar ve içerikler dikkatle değerlendirilerek, etiketleme işlemi her bir kümenin anlamını en iyi şekilde yansıtacak biçimde gerçekleştirilmiştir. BERT (Bidirectional Encoder Representations from Transformers) modeli hazırlanmış veri seti ile eğitilmiştir Eğitilen model yardımıyla fenomenin gönderileri sınıflandırılarak hitap ettiği kitle yüksek doğruluk oranı ile tespit edilebilmektedir.

Bu çalışma, sosyal medya kullanımının hızla arttığı bu dönemde, Türkçe dili için sosyal medya analitiği ve metin sınıflandırma alanında önemli bir boşluğu doldurmaktadır. Çalışmanın katkıları iki ana başlıkta değerlendirilebilir: Türkçe metinlerin BERTopic mimarisiyle sınıflandırılması, Türkçe dilinde daha fazla araştırma ve uygulamayı teşvik ederek akademik bir katkı sunmaktadır. Ayrıca, elde edilen sınıflandırma başarısıyla fenomen-ürün hedef kitle eşleşmesini mümkün kılarak sosyal medya pazarlamasına ticari bir katkı sağlamaktadır.

2. Literatür Taraması (Related Work)

Bu bölümde öncelikle literatürde yer alan farklı modeller ile başarı elde edilmiş metin sınıflandırma çalışmalarına yer verilmiştir. Daha sonra BERT modeli ile yapılan metin sınıflandırma çalışmaları ile devam edilmiş olup son bölümde ise sosyal medya uygulamalarına yapılan metin sınıflandırma çalışmalarının detaylarına yer verilmiştir.

Literatürde metin sınıflandırma için birçok yöntem ve uygulama alanı bulunmaktadır. Örneğin Türkçe haber metinlerinin sınıflandırılması için Destek Vektör Makinesi, Rastgele Orman ve Naive Bayes sınıflandırma algoritmalarını karşılaştıran çalışmada, 4.900 satırlık haber metinlerinden oluşan veri seti 7 kategoriye ayrılmıştır. İşlemler sonucunda %91 doğruluk oranı ile Naive Bayes algoritması diğer algoritmalara göre en başarılı performansı göstermiştir (Uslu ve Özmen-Akyol, 2021).

Bir internet sitesinin e-ticaret sitesi olup olmadığına karar veren bir uygulama için ön işleme aşamaları gerçekleştirilerek etiketlenen 273 adet site verisi K-En Yakın Komşu ve Naive Bayes algoritmaları ile eğitilmiş, diğer algoritmalara göre Naive Bayes algoritmasının daha iyi sonuç verdiği görülmüştür (Kaşıkçı ve Gökçen, 2019).

Günümüzde metin sınıflandırmada klasik makine öğrenmesi yöntemleri yerine büyük veri setleri ve karmaşık görevler için daha uygun olan derin öğrenme yöntemleri daha fazla kullanılmaktadır. Türkçe haber metinlerinin sınıflandırılması için Konvolüsyonel Sinir Ağları ve Word2Vec metodu kullanılarak yapılan metin sınıflandırma çalışmasında, klasik makine öğrenmesi sınıflandırma algoritmalarından daha iyi bir performans (%93,3 doğruluk) elde edildiği belirtilmiştir. (Acı ve Çırak, 2019).

10.517 e-postadan oluşan veri seti ile, alınan e-postaları önemine göre sınıflandırmak için Word2Vec algoritması kullanılmıştır. 200 e-postadan oluşan test verisi ile sistem başarısı test edilmiş ve %91 oranında doğruluk ile başarı elde edildiği ifade edilmiştir (Sel ve Hanbay, 2019).

Türkçe dilinde yazılan bilimsel metinlerin sınıflandırılması ile ilgili gerçekleştirilen çalışmada önceden eğitilmiş Türkçe bir BERT modeli üzerinde ince ayar yapılmış ve model %96 doğruluk oranı göstermiştir (Özkan ve Kar, 2022).

BERT modeli ve geleneksel makine öğrenmesi modelleri kullanılarak dört farklı deney ile metin sınıflandırma yapılmıştır. Çalışmada sosyal medyada paylaşılan iletiler, film dizi eleştirileri, haber içerikleri gibi veri setleri üzerinde karşılaştırmalı analizler gerçekleştirilerek sonuçlar sunulmuş BERT modelinin diğer modellere göre başarı gösterdiği görülmüştür (González-Carvajal ve Garrido-Merchán, 2020).

Web sitesi URL'lerinden çıkarılan metinler üzerinde önceden eğitilmiş BERT modeli ile sınıflandırılan çalışmada %98 doğruluk ve %67 F1 skoru elde edildiği belirtilmiştir. (Çepni vd., 2023).

Sosyal medya fenomenlerini ve gönderilerini; Naive Bayes, K-En Yakın Komşu, Destek Vektör, Rastgele Orman ve BERT modelleri ile sınıflandırarak karşılaştırılan çalışmada, BERT modelinin diğer modellere göre fenomenleri %98, gönderilerini ise %96 doğruluk ile daha başarılı bir şekilde sınıflandırdığı belirtilmiştir (Kim vd., 2020).

Instagram yorumlarını otomatik olarak sınıflandıran sistem için Destek Vektör Makineleri (Support Vector Machine kısaca SVM) ve Evrişimli Sinir Ağı (Convolutional Neural Network kısaca CNN) algoritmaları karşılaştırılmıştır. %84,23 doğruluk oranı ile CNN algoritmasının daha iyi sonuç verdiği belirtilmiştir (Prabowo ve Purwarianti, 2017).

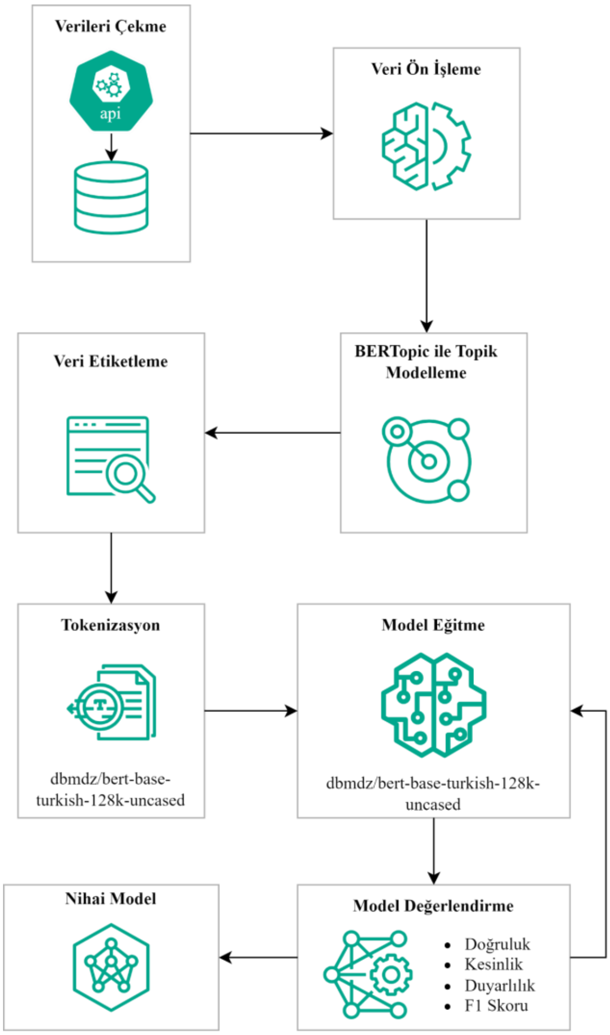
Yapılan literatür taramasında, metin sınıflandırma alanında çeşitli algoritmalar ve derin öğrenme modellerinin başarıları vurgulanmıştır. Bu çalışmalar incelendiğinde klasik makine öğrenmesi yöntemlerinin ve GaussianNB (Gaussian Naive Bayes), K-En Yakın Komşu (K-Nearest Neighbors kısaca KNN), Destek Vektör Sınıflandırması (Support Vector Classifier kısaca SVC) ve Rastgele Orman (Random Forest) gibi yaygın olarak kullanılan sınıflandırma modelleri kullanılarak geliştirilen modellerde en fazla %40.70 başarı oranı elde edildiği gözlemlenmiştir ve bu modellerin çalışmamız için yeterli doğruluk oranlarını veremeyeceği sonucu çıkarılmıştır. Özellikle BERT modeli, Türkçe metin sınıflandırılmasında yüksek doğruluk oranları elde ederek literatürde öne çıkmıştır. Literatürde BERT modeli ile yapılmış sınıflandırma çalışmalarının doğruluk oranları yapılan çalışmanın başarıya ulaşmasına referans olmuştur. Çalışmamız, BERT modelinin sosyal medya analizleri ve metin sınıflandırma alanındaki uygulamalarına yeni bir perspektif kazandırarak literatüre önemli bir katkı sağlamaktadır.

Literatürde Türkçe dili için metin sınıflandırma çalışmalarının sayısı oldukça sınırlıdır. BERTurk ile yapılan bu çalışmada, yalnızca gönderi açıklama metinleri kullanılarak %92 doğruluk değeri elde edilmiştir. Bu sonuç, benzer çalışmalarda (Kim vd., 2020) rapor edilen %60 doğruluk değerini önemli ölçüde aşmaktadır. Dolayısıyla, bu çalışma hem BERTurk'ün etkinliğini vurgulamakta hem de gönderi açıklama metinlerinin derin öğrenme modelleri için güçlü bir veri kaynağı olabileceğini ortaya koymaktadır.

Türkçe dilinde metin sınıflandırma çalışmaları ve fenomen-ürün hedef kitle eşleştirmesi konusunda literatürde makine öğrenmesi yöntemleri ile yapılmış çalışmaların yeterli seviyede olmaması bu alanda yapılacak çalışmalara duyulan ihtiyacı göstermektedir. Çalışmamız, Türkçe dilinde yapılacak metin sınıflandırma çalışmaları için referans oluşturmaktadır. Sosyal medya pazarlamasında doğru fenomen-hedef kitle eşleşmesini sağlayarak, markaların daha etkili kampanyalar oluşturmasına olanak tanımaktadır. Literatüre yenilikçi bir yaklaşım sunmakta olan bu çalışma, ilgili amaçla yapılacak gelecekteki çalışmalara önemli bir referans oluşturmaktadır.

3. Materyal ve Metod (Material and Method)

Bu bölümde çalışma kapsamında kullanılan yöntemler açıklanmıştır. Çalışmada kullanılan metodoloji Şekil 1'de verilmiştir.



Şekil 1. Metodoloji

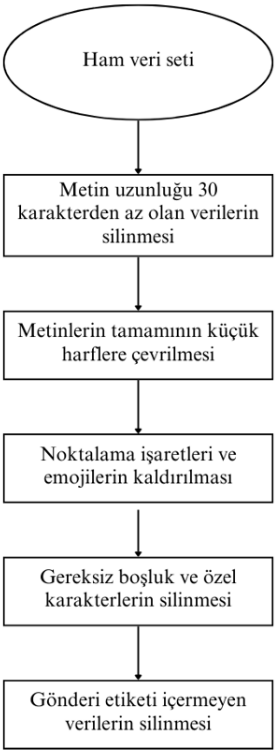
3.1 Veri Toplama

Çalışma kapsamında kullanılacak olan veri setini oluşturmak için, kişisel verilerin ihlalini önlemek adına Instagram hesapları halka açık olan fenomenlerin gönderileri elde edilmiştir. Instagram hesapları halka açık olan fenomenlerin gönderileri, Apify adlı üçüncü taraf hizmeti olan veri kazıma (web scraping) platformu kullanılmıştır. Apify, kullanıcıların e-ticaret siteleri, haber siteleri, sosyal medya platformları gibi çeşitli web sitelerinden veri çekmelerine olanak tanıyarak bu verilerin çeşitli amaçlar doğrultusunda kullanılabilmesini mümkün kılar (Apify, 2022). Bu çalışma kapsamında Apify, Instagram üzerinden 1005 farklı fenomenin gönderilerinin çekilmesinde kullanılmıştır. Böylelikle çalışma kapsamında elde edilen veri seti, güvenilirlik ve etik kurallara uygunluk çerçevesinde oluşturulmuştur.

Bu yöntemle toplamda 1.005 farklı fenomenin 647.951 adet gönderi verisi elde edilmiştir. Veri ön işleme adımında sonra analiz için toplamda 610.600 gönderi verisi, çalışmada kullanılan nihai veri setini oluşturmuştur.

3.2 Veri Ön İşleme

Elde edilen veri setine, model eğitimi ve analizler için veri ön işleme adımı uygulanmıştır. Bu adımdaki amaç, model performansını maksimize etmek amacıyla daha kaliteli bir veri seti elde etmektir. Bu bağlamda, fenomen gönderilerinin açıklama metinleri işlenmiştir. Uygulanan adımlar Şekil 2'de verilmiştir.


Ham veri seti
Metin uzunluğu 30 karakterden az olan verilerin silinmesi
Metinlerin tamamının küçük harflere çevrilmesi
Noktalama işaretleri ve emojilerin kaldırılması
Gereksiz boşluk ve özel karakterlerin silinmesi
Gönderi etiketi içermeyen verilerin silinmesi

Şekil 2. Veri ön işleme adımları

Öncelikle, açıklama metin uzunluğu 30 karakterden kısa olan veriler, veri setinden çıkarılmıştır. Türkçe dilinde ortalama kelime uzunluğu göz önüne alındığında, 30 karakter genellikle yaklaşık 4 ila 5 (Dalkılıç vd., 2003) kelimeye denk gelmektedir. Bu nedenle, 30 karakterden kısa metinlerin veri setinden çıkarılması, yeterli bilgi ve bağlam sağlayan daha anlamlı açıklamaların analiz edilmesine olanak tanımaktadır. Bu seçim, metinlerin yeterli bilgi içermesini sağlamak ve analizde daha anlamlı sonuçlar elde etmek amacıyla yapılmıştır. Daha sonra, açıklama metinlerinde herhangi bir büyük harf olmaması adına metinlerin tamamı küçük harflere dönüştürülmüştür çünkü kullanılacak model büyük-küçük harf hassasiyeti taşımamaktadır. Bu dönüşüm, metinlerdeki büyük ve küçük harf farklılıklarını ortadan kaldırarak, modelin tüm metinleri aynı şekilde değerlendirmesini ve karşılaştırmasını sağlar. Bu nedenle, büyük harflerin küçük harflere dönüştürülmesi, modelin doğruluğunu ve işlem sürecini iyileştirmek adına uygulanmıştır. Noktalama işaretleri/emojiler kaldırılmıştır. Bu adım, metinlerin tutarlılığını artırmak ve dil modelinin sadece anlamlı kelimelere odaklanmasını sağlamak için önemlidir. Noktalama işaretleri ve emojiler, modelin analizini karmaşıklaştırabileceğinden, bunların temizlenmesi gereklidir. Ek olarak gereksiz boşluklar ve özel karakterler de temizlenerek tüm açıklama metinleri aynı formata getirilmiştir. Sonrasında, gönderi etiketi (hashtag) içermeyen gönderiler veri setinden çıkarılmıştır. Çünkü sosyal medya üzerinden ürün veya hizmet tanıtımı yapan fenomenler, ürün veya hizmetin reklam olduğunu belirtmek zorundadır ve aynı zamanda reklamı yapılan gönderinin daha fazla kişiye ulaşması amacıyla gönderi etiketi kullanma eğilimindedirler. Bu nedenle, gönderi etiketi içermeyen verilerin veri setinden çıkarılması, reklam hedef kitlesini daha doğru bir şekilde belirlemede önemli bir rol oynamaktadır. Bu adım, veri setinin kalitesini artırmaya ve sonuçların doğruluğunu artırmaya yönelik bir önlem olarak uygulanmıştır.

3.3 Veri Etiketleme

Veri ön işleme adımından sonra elde edilen nihai veri seti, fenomen gönderilerinin açıklamalarını baz alarak verileri etiketlemek amacıyla BERTopic modeli ile kümelenmiştir.

BERTopic mimarisi, doğal dil işleme alanında yaygın olarak kullanılan metin konularının temsillerini (topic) tespit etmek ve bu konuları kümeler halinde gruplamak için kullanılan BERT dil modeli mimarisine dayalı bir modeldir (Grootendorst, 2022).

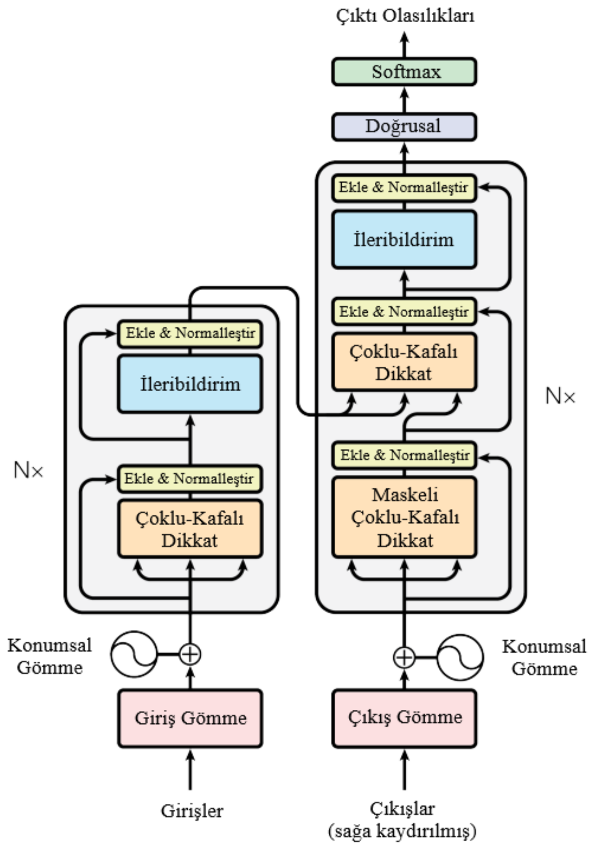
3.4 Mimari ve Model

Çalışma kapsamında iki farklı amaçla iki farklı dönüştürücü (transformer) mimarisi kullanılmıştır;

1. Binlerce satır veriden oluşan veri setini önceden belirlenmiş kategoriler doğrultusunda etiketlemek için BERTopic mimarisi kullanılmıştır.
2. Veri seti etiketleri elde edildikten sonra sınıflandırma modelinin fenomenlerin hitap ettiği hedef kitle doğrultusunda sınıflandırabilmek için BERT mimarisi kullanılmıştır.

3.4.1 Dönüştürücü Mimarisi (Transformer Architecture)

Dönüştürücüler, doğal dil işleme ve diğer sıralı veri işleme görevlerinde kullanılan bir sinir ağı mimarisidir. Özellikle, uzun mesafe bağımlılıkları ele almak ve büyük veri kümeleri üzerinde paralel işlem yapmak için etkilidir. Bu mimari, dikkat mekanizmasını (attention mechanism) içeren bir yapıya sahiptir ve daha önceki dil modellerinden önemli ölçüde farklılık gösterir (Vaswani vd., 2017). Dönüştürücü mimarisi, birçok tekrarlayan katman içerir ve her bir katman, birbiriyle bağlantılıdır. Her katman, dikkat mekanizmasını kullanarak girdi verilerini işler. Dikkat mekanizması, her bir girdi öğesinin, diğer tüm öğelerle olan ilişkisini hesaplar ve bu ilişkilere göre ağırlıklar atar. Bu sayede, her bir öğenin önemi belirlenir ve dikkate alınır. Dönüştürücü mimarisi, genellikle bir kodlayıcı (encoder) ve bir çözücü (decoder) olarak iki ana bileşenden oluşur (Aitken vd., 2021). Kodlayıcı, girdi verilerini temsil eden vektörler oluştururken, çözücü, bu vektörleri hedef çıktılara dönüştürür. Her bir bileşen, birçok tekrarlayan katmana sahiptir ve her katman, birden fazla dikkat mekanizması içerir. Dikkat mekanizması, girdi vektörlerinin birbiriyle olan ilişkilerini hesaplar. Her bir girdi vektörü, diğer tüm vektörlere olan benzerliklerine göre ağırlıklar alır. Bu ağırlıklar, her bir vektörün diğerleri üzerindeki etkisini belirler. Özellikle, uzun mesafe bağımlılıkları ele almak için etkilidir ve önceki dil modellerinden daha iyi sonuçlar verir. Dönüştürücü mimarisi, dil modelleri ve diğer sıralı veri işleme görevlerinde genellikle kullanılır. Büyük metin veri kümeleri üzerinden eğitilmiş olan modeller, genellikle az miktarda etiketlenmiş veri ile yüksek doğruluk sağlar.

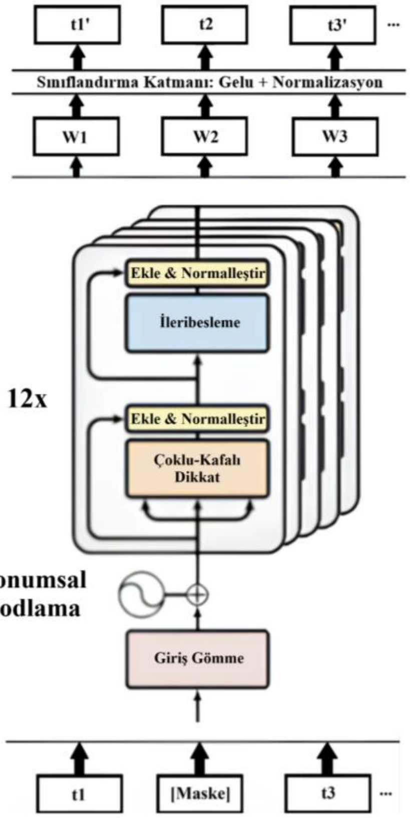


Şekil 3. Dönüştürücü - model mimarisi (Vaswani vd., 2017)

BERT, dönüştürücü mimarisi ile geliştirilmiştir ve sadece kodlayıcı kısmını kullanarak metni çift yönlü (hem ileri hem de geri yönde) analiz eden bir dil dönüştürücü modelidir. Dönüştürücü model mimarisi Şekil 3'te verilmiştir.

3.4.2 BERT Mimarisi

BERT, Google tarafından geliştirilen ve doğal dil işleme alanında devrim niteliğinde bir ilerleme olarak kabul edilen bir modeldir. Bu model, büyük miktarda metin verisi üzerinde ön eğitilmiş bir dil modelidir ve sıralı veri işleme için son derece etkilidir. BERT, hem sol hem sağ bağlamı dikkate alan bir biçimde kelimeleri bir araya getirir. Bu, metin içerisindeki her kelimenin anlamını, hem önceki hem de sonraki kelimelerin bağlamından elde eder. Bu şekilde, metnin daha geniş bir bağlamını anlayabilir ve daha derin bir semantik anlam çıkarabilir.



Şekil 4. On iki kodlayıcı bloğa sahip, dönüştürücü tabanlı BERT temel mimarisi (Khalid vd., 2021)

Şekil 4'te on iki kodlayıcı bloğa sahip, dönüştürücü tabanlı BERT temel mimarisi verilmiştir. BERT, sadece kodlayıcı kısmını içeren bir dil modelidir. Bu, dönüştürücü mimarisinin sadece kodlayıcı bileşenini içerdiği anlamına gelir. Kodlayıcı, girdi verilerini temsil eden vektörler oluşturur, ancak bu vektörlerin nasıl kullanılacağı veya çözümleneceği konusunda herhangi bir bilgi bulunmaz. Bu özellik, BERT'in önceden eğitilmiş bir dil modeli olarak kullanılmasını sağlar. BERT, büyük metin veri kümeleri üzerinde eğitimli olduğu için, genel dil yapısını ve anlamını öğrenir. Ancak, belirli bir görev için kullanılmak üzere eğitilmesi gerekebilir. Örnek olarak, sınıflandırma görevleri için, BERT modelinin kodlayıcı kısmı, sınıflandırma modeliyle birleştirilerek yeniden eğitilir ve öğrenilmiş temsiller kullanılarak sınıflandırma yapılır. Bu nedenle, BERT modeli, kodlayıcı bileşeninin özelliklerinden yararlanarak çeşitli doğal dil işleme görevleri için kullanılabilir. Kodlayıcı, metin verilerini temsil eden vektörler oluştururken, bu vektörlerin çözümlenmesi veya belirli bir görev için kullanılması, modelin yeniden eğitilmesini gerektirir (Devlin vd., 2018).

3.4.3 BERTopic Mimarisi

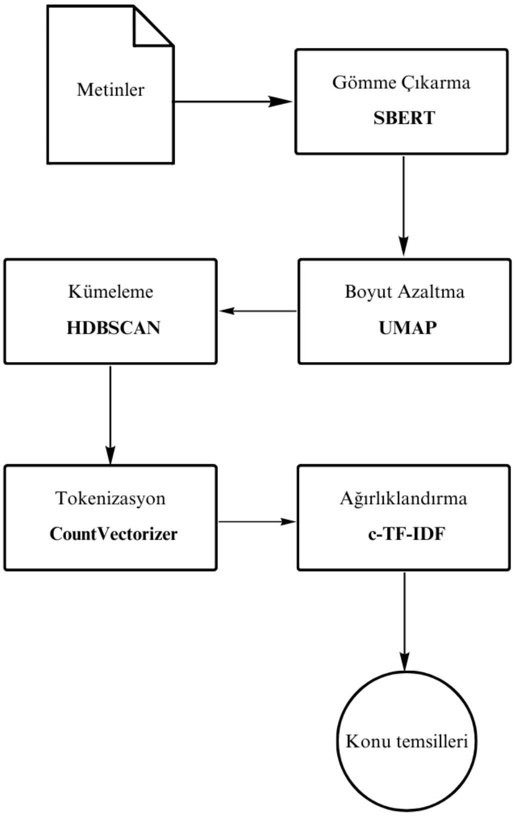
BERTopic modeli, BERT ve Sınıf tabanlı kelime frekansı–ters doküman frekansı (Class-based term frequency- inverse document frequency kısaca c-TF-IDF) tekniklerini kullanarak metin konularını tespit edip kümeleyen bir konu modelleme tekniğidir (Grootendorst, 2022). Temel olarak metinlerden cümle gömmelerini elde ettikten sonra benzer cümleleri kümeleyerek kümeleri oluşturur. Şekil 5'te BERTopic modeli ile küme oluşturma adımları verilmiştir. Kümeler; gömme çıkarma, boyut azaltma, kümeleme, tokenizasyon ve ağırlıklandırma olmak üzere beş temel adımdan geçerek oluşturulmaktadır.

BERTopic, metinlerden cümle gömmelerini elde etmek için Cümlü Dönüştürücü (Sentence-BERT kısaca SBERT) tekniğini kullanmaktadır. SBERT mimarisi, BERT mimarisinin geliştirilmiş versiyonudur. BERT mimarisi cümleleri kelime kelime işlerken SBERT cümleleri tek seferde işlediği için metin semantiklerini daha kapsamlı bir şekilde anlayabilir (Reimers ve Gurevych, 2019).

Kümeleme adımında yüksek boyutlu gömme vektörlerinin kümelenmesi daha zor ve komplike olduğundan, kümeleme adımından önce Tekdüze Manifold Yaklaşımı ve Yansıtması (Uniform Manifold Approximation and Projection kısaca UMAP) kullanılarak cümle gömmelerinin boyutları azaltılır.

Cümle gömmelerinin boyutu azaltıldıktan sonra yoğunluk temelli bir kümeleme algoritması olan Gürültülü Uygulamalar için Hiyerarşik Yoğunluk Tabanlı Uzamsal Kümeleme (Hierarchical Density-Based Spatial Clustering of Applications with Noise kısaca HDBSCAN) algoritması ile benzer metin gömmeleri kümelenir.

BERTopic mimarisi, tokenizasyon adımında ise doğal dil işleme alanında yaygın olarak kullanılan kelimelerin metinde geçme sıklığı ve metin içerisindeki önemini dikkate alarak tokenizasyon işlemi için Sayı Vektörleştirici (CountVectorizer) metin işleme tekniğini kullanır.



Şekil 5. BERTopic modeli ile kümeleme adımları

Sonraki adımda ise elde edilen tokenler, c-TF-IDF tekniği ile kümeleri oluşturulur. Kelime frekansı–ters doküman frekansı (Term frequency- inverse document frequency kısaca TF-IDF) tekniği, metin belgelerini kelimenin alaka düzeyine göre vektörleştirirken, c-TF-IDF aynı işlemi tek bir kategorideki tüm belgeleri tek bir belge olarak ele alır ve yapar. Böylelikle ilgili küme özelinde kümeyi en iyi şekilde temsil eden kelimeler elde edilmiş olur (Liu vd., 2018).

3.5 Model Eğitme

Bu bölümde, model eğitimi aşamasında kullanılan model tanıtılmıştır.

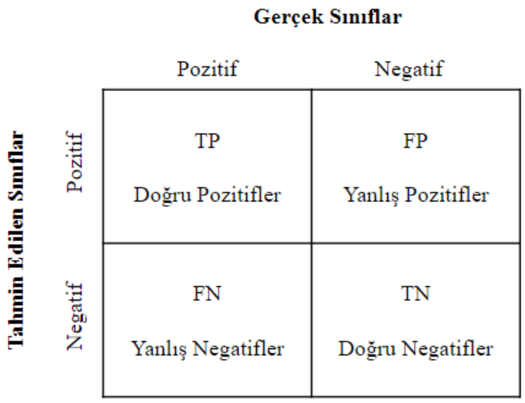
3.5.1 BERTurk ile Sınıflandırma

Sıralı veri işleme için BERTurk modelinin kodlayıcı (encoder) kısmını kullanır. Bu kodlayıcı, girdi metin dizisini bir dizi temsil vektörüne dönüştürür. Bu vektörler, girdi metnin anlamını ve bağlamını yansıtır. Ardından, bu temsil vektörleri, bir sınıflandırma katmanına beslenir. Sınıflandırma katmanı, bu temsil vektörlerini alır ve belirli sınıflara ait olasılık değerlerini tahmin eder. Bu, tipik olarak bir "softmax" aktivasyon fonksiyonu ile gerçekleştirilir.

3.6 Model Performans Değerlendirmesi

Sınıflandırma model performansı karışıklık matrisi (confusion matrix), doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve f1 skoru (f1-score) performans metrikleri ile değerlendirilmiştir.

Karışıklık matrisi, sınıflandırma modelinin performansını değerlendirmek için kullanılan, model tahminlerini özetleyen bir performans değerlendirme metriğidir. Sınıflandırma modeli performansı değerlendirilirken modelin yaptığı hataları ve zayıflıkları hakkında daha kapsamlı çıkarımlar yapılmasına yardımcı olur. Karışıklık matrisi Şekil 6'daki gibi ifade edilir.



		Gerçek Sınıflar	
		Pozitif	Negatif
Tahmin Edilen Sınıflar	Pozitif	TP Doğru Pozitifler	FP Yanlış Pozitifler
	Negatif	FN Yanlış Negatifler	TN Doğru Negatifler

Şekil 6. Karışıklık matrisi

Karışıklık matrisinde doğru pozitifler doğru şekilde sınıflandırılan pozitif örnekleri, yanlış pozitifler yanlış şekilde sınıflandırılan negatif örnekleri, yanlış negatifler yanlış şekilde sınıflandırılan pozitif örnekleri, doğru negatifler ise doğru şekilde sınıflandırılan negatif örnekleri belirtmektedir.

Bir sınıflandırma probleminde doğruluk metriği, modelin ne kadar örneği doğru tahmin ettiğini ifade eder ve aşağıdaki gibi hesaplanır;

$$Doğruluk \; = \frac{TP + TN}{TP + TN + FP + FN} \qquad (1)$$

Kesinlik metriği, bir sınıflandırma modelinin pozitif olarak tahminlediği örneklerin kaç adetinin gerçekten pozitif olduğunu belirtir ve aşağıdaki gibi hesaplanır;

$$Kesinlik \; = \frac{TP}{TP + FP} \qquad (2)$$

Duyarlılık metriği ise sınıflandırma modelinin pozitif olarak tahmin etmesi gereken sınıfların ne kadarını pozitif olarak tahmin ettiğini belirtir ve aşağıdaki gibi hesaplanır;

$$Duyarlılık \; = \frac{TP}{TP + FN} \qquad (3)$$

F1 skoru ise kesinlik ve duyarlılık metriklerinin harmonik ortalaması alınarak hesaplanır ve uç durumların göz ardı edilmemesini, daha doğru analizler yapılabilmesini sağlar. F1 skoru aşağıdaki gibi hesaplanır;

$$F1\,Skoru \; = 2\,x\,\frac{Kesinlik\;x\;Duyarlılık}{Kesinlik + Duyarlılık} \qquad (4)$$

4. Değerlendirme (Evaluation)

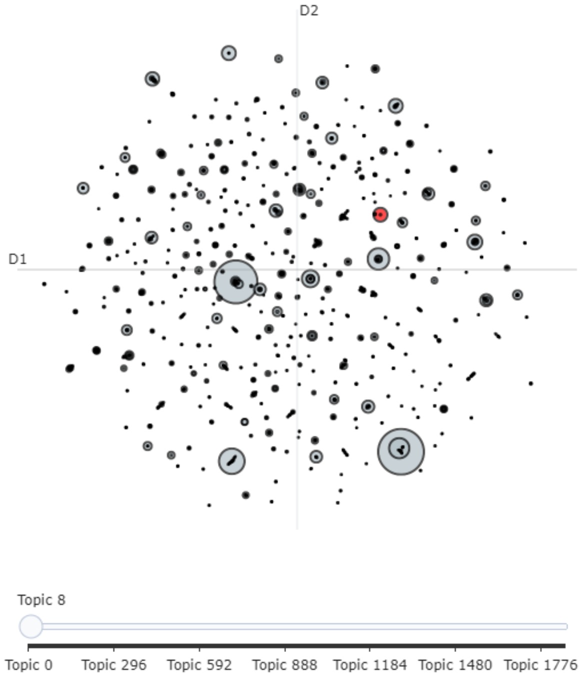
Bu çalışma, fenomenleri Instagram'daki gönderilerinde bulunan açıklamalar doğrultusunda hitap ettiği kategoriyi tanımlayabilmek adına doğal dil işleme teknikleriyle sınıflandırmayı amaçlamaktadır. Bu amaç doğrultusunda öncelikle çalışma kapsamında kullanılan veri seti, Apify ile Instagram hesabı halka açık olan fenomenlerin gönderi bilgilerinin çekilmesi ile oluşturulmuş ve veriyi doğal dil işleme model eğitimine hazırlamak adına veri ön işleme gerçekleştirilmiştir.

Elde edilen veri seti ile BERTopic modelini kullanarak gönderi açıklamalarının kümeleri (topic) oluşturulmuştur. Toplamda 1862 adet farklı küme oluşmuştur. BERTopic modeli tarafından belirli bir küme ile ilişkilendirilemeyen verileri temsil eden "-1" numaralı küme görmezden gelinmiştir. Kümeye ait anahtar kelimeler ve örnek verilerin detaylı incelenmesi ile kümeler önceden belirlenmiş olan 18 kategoriye göre etiketlenmiştir. Tablo 1'de, örnek olması için model tarafından oluşturulan 21 kümenin etiketlenmiş hali, bu kümelere ait veri setindeki örnek sayısı ve kümeye ait anahtar kelimeler verilmiştir.

Tablo 1. Model tarafından oluşturulan ilk 21 kümenin etiketlenmiş hali

Etiket	Kümeler	Örnek Sayısı	Kümelere Ait Anahtar Kelimeler
-	0	21524	seni, sen, ki, hep, ne, şey, anne, diye, hayat, sana
Yiyecek ve İçecek	1	19078	kaşığı, bardağı, yemek, yumurta, su, ekleyip, gr, adet, paket, şeker
Seyahat ve Ulaşım	2	6826	seyahat, travel, gezi, şehir, tatil, antik, deniz, yer, burası, gezgin
Yiyecek ve İçecek	3	4795	kahve, coffee, coffeetime, coffeerem, coffeelover, coffeeoftheday, kahvekeyfi, kahvesi,coffeeculture, coffeeshots
Kitaplar ve Edebiyat	4	4161	kitap, kitabı, kitap önerisi, roman, edebiyat, kitapkurdu, kitaptavsiyesi, kitaplar, kitapları, neokuyorum
Otomobiller ve Araçlar	5	3065	otomobil, otomobiltutkusu, suv, porsche, bmw, arabatutkusu, otoparkcom, ford, arabasevdası, elektrikli
Güzellik ve Egzersiz	6	2229	diyet, kilo, diyetisyen, beslenme, onlinediyet, diyetyemekleri, zayıflama, roket, diyeti, diyetisyenpinardemirkaya
Güzellik ve Egzersiz	7	2051	makeup, makyaj, lipstick, makeuptutorial, ruj, maccosmeticsturkiye, maybelline, matte, makyajı, eyeliner
Güzellik ve Egzersiz	8	2024	saç, saçlarımı, hair, hairstyle, saçlar, saçlarım, saçları, şampuan, bakım, saçbakımı
Güzellik ve Egzersiz	9	1956	cilt, cildi, bakım, serum, cildin, nemlendirici, kremi, ciltbakımı, asit, cilde
Çevrimiçi Topluluklar	10	1917	reelsinstagram, iphone, reels, reelsvideo, instagram, samsung, pro, reelsindia, reelesviral, apple
Çevrimiçi Topluluklar	11	1593	youtube, video, videonun, videoyu, kanalımda, yayında, youtubeda, kanalda, abone, videosu
Güzellik ve Egzersiz	12	1555	gununegzersizi, bacak, egzersiz, kalça, core, kaslarını, egzersizler, hareket, çalıştıran, omuz
İşler ve Eğitim	13	1483	tongucakademi, ogrenci, lgs, öğrenci, okul, ders, teog, öğretmenler, öğretmenlerimizin, seyev
Seyahat ve Ulaşım	14	1402	kamp, kampalani, kampvedogahayati, kampvedogadakiler, camping, kamptagram, kampmudavimleri, kampturkiye, karavan, yolacikyolacik
Haberler	15	1390	galatasaray, futbol, football, nba, fenerbahçe, maç, süperlig, messi, gol, transfer
Güzellik ve Egzersiz	16	1386	modanisa, renk, indirim, far, kodu, rengi, paleti, renkler, palet, makyaj
Ev ve Bahçe	17	1383	dekorasyon, bekliyorumsayfamızı, interiordesign, dekorasyonönerileri, yorumlarınızı, türkkahvesikeyfi, sunumvetarif, interior, balkon
Alışveriş	18	1368	hediye, çekiliş, cekilis, arkadaşınızı, etiketlemek, kişiye, çekillişvar, yapmanız, gerekenler, ceikilisvar
Güzellik ve Egzersiz	19	1357	moda, fashion, outfitoftheday, outfit, elbise, dress, ootd, kombin, style, ekinde
İnsan ve Toplum	20	1242	gelisimadam, girisimcilik, startup, fazlası, internettenparakazanmak, sosyalmedyayonetimi, makemoney, etiketlesosyalmedyauzmani, girisim, gelisim

Model tarafından oluşturulan tüm kümelere ait mesafe haritası Şekil 7 üzerinde verilmiştir. Örnek olarak kırmızı renkle belirtilen kümeye ait anahtar kelimeler "saç, saçlarımı, hair, hairstyle, saçlar, saçlarım, saçları, şampuan, bakım, saçbakımı" olarak listelenmiştir. Yanında bulunan diğer kümeye ait anahtar kelimeler ise "cilt, cildi, bakım, serum, cildin, nemlendirici, kremi, ciltbakımı, asit, cilde" olarak belirlenmiştir. Bu iki kümenin birbirine yakın olmasının nedeni, her iki kümenin de kişisel bakım ürünleri ile ilgili olmasıdır. Saç ve cilt bakımı, kişisel bakım kategorisi altında ortak bir ilgi alanı oluşturur, bu da kümelerin yakınlığını açıklar.



Şekil 7. Kümeler arası mesafe haritası

Başlangıçta veri çekme aşamasında fenomenlerin tüm gönderileri çekilmiştir. Bu gönderiler arasında, ürün veya hizmet tanıtımı içermeyen bazı kişisel gönderiler de bulunmuştur. Örneğin, kişinin kişisel hayatına dair bilgiler ve tatil paylaşımları gibi gönderiler kişisel içeriklere örnek teşkil etmektedir. Veri etiketleme aşamasında bu gönderiler veri setinden çıkarılmıştır. Bu sayede sadece reklam içeren gönderilerden oluşan bir veri seti elde edilmiştir. Kümelere ait anahtar kelimeler incelendiğinde, örnek sayısının azalmasıyla birlikte bu kelimeler arasındaki benzerliğin de azaldığı gözlemlenmiştir. Bu sebeple, kümelere ait örnek sayıları için iki farklı eşik değeri (threshold) belirlenmiştir: Bu değerler 190 ve 140 olarak belirlenmiştir. Belirlenen bu sınır değerleri, her bir kümenin temsil ettiği anahtar kelimelerin içeriklerine dayanmaktadır. Örneğin, 185 örneğe sahip bir kümenin anahtar kelimeleri ("zafer, vatan, şehitlerimizi, ağustos, tayyare, minnetle, oy, zaferbayramı, kutlu, anıyoruz") belirli bir temayı net bir şekilde temsil etmesine rağmen, reklam ile ilgili belirlediğimiz 18 kategoriye tam olarak uymamaktadır. Bu nedenle bu tür kümeler yoğun veri setine dahil edilmemiştir. Benzer şekilde, 178 örneğe sahip bir küme, ("amigurumitarifleri, kisiyeozelhediye, dogumgunuhediyesi, amigurumiteknikleri, amigurimitarifleri, freetarif, crocheting, dogumgunu, elemegi, loveislove") anahtar kelimelerini içermekte ve bu kelimeler kısmen “yiyecek ve içecek” kategorisine denk gelmektedir. Ancak bu küme, “yiyecek ve içecek” kategorisi dışındaki kategoriler ile de örtüşebileceği için yoğun veri setine dahil edilmemiş, ancak orta yoğun veri setine dahil edilmiştir. Orta yoğun veri seti için 140 sınırının belirlenmesindeki temel neden ise bu noktada kümelerin 2 kategoriyi de içermeye başlamasıdır. 140’ın altında, örneğin 136 örneğe sahip bir kümenin anahtar kelimeleri ("flowers, çiçekler, tongucakademi, boardmasters, repostapp, tene, tonguçlamaya, teog, flower") “işler ve eğitim”, “ev ve bahçe” ve “hobiler ve boş zaman uğraşları” gibi 2’den fazla kategoriye ait veriler içermekte olup, veri setinin dengesini bozabileceğinden bu küme herhangi bir veri setine dahil edilmemiştir. Kümelere ait örnekler incelendiğinde 140-190 aralığında örneğe sahip olan kümelerin 1 veya maksimum 2 kategoriyi temsil ettiği, 140’dan az örnek bulunan kümelerin ise en az 2 veya daha fazla kategoriyi temsil ettiği görülmüştür.

Sonuç olarak 1 kategoriyi temsil eden kümeleri içerdiği için 190 ve üzeri örneğe sahip kümeler etiketlenerek yoğun veri setine dahil edilmiş, 1 veya maksimum 2 kategoriye ait verileri içerdiği için 140 üzeri örneğe sahip kümeler ise etiketlenerek orta yoğun veri setine dahil edilmiştir. Bu kriterler, daha homojen ve belirli kategorilere net bir şekilde uyan veri setleri oluşturmak amacıyla belirlenmiştir. Oluşturulan veri setlerine ait veri miktarları Tablo 2’de gösterilmiştir.

Tablo 2. Veri setlerine ait veri miktarları

Veri Seti	Eğitim	Doğrulama	Test
Orta Yoğun Veri Seti	124,726	14,506	14,507
Yoğun Veri Seti	85,666	10,111	10,113

Veri setlerine, Türkçe metinler ile ön-eğitimli olan BERTurk modelleri arasından “dbmdz/bert-base-turkish-128k-uncased” modeli ile tokenizasyon uygulanmıştır (tokenizing). Kullanılan BERTurk modeli diğer modellere kıyasla daha büyük ve daha performanslıdır. Türkçe tıbbi metin sınıflandırmasında BERTurk modelinin üstün performansı, yapılan bir çalışmada 0.93 F-skoru ile kanıtlanmıştır; bu, çok dilli BERT modelinin 0.82 F-skoruna kıyasla önemli ölçüde daha yüksektir (Celikten vd., 2021). Tokenizasyon işlemi ile metin verileri sayısal vektör temsillerine dönüştürülerek modelin anlayabileceği formata getirilmiştir.

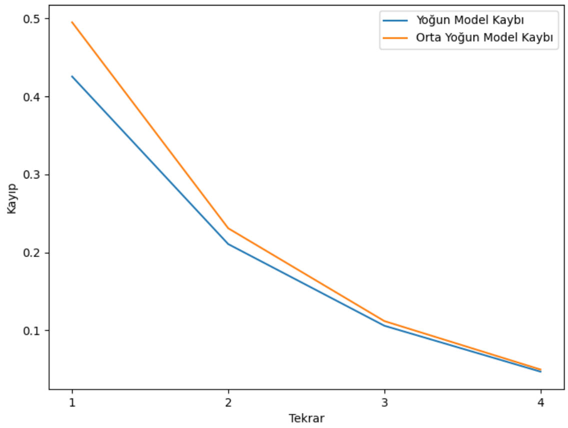
Tokenizasyon işlemi uygulanmış olan veri setleri, modelin eğitimi ve performansının değerlendirilmesi için %80 eğitim, %10 doğrulama ve %10 test verisi olarak bölünmüştür. Bu bölme işlemi, modelin genel bir performans ölçütü elde etmesine olanak tanırken, aynı zamanda modelin aşırı uyum (overfitting) gibi sorunlara karşı direncini test etmek için de gereklidir.

Eğitim aşamasında, tokenizasyon işleminden geçmiş veriler BERTurk modeli ile eğitilmiştir. Tablo 3‘te verilmiş olan parametreler ile eğitim gerçekleştirilmiştir. Bu parametreler, modelin eğitim verisi üzerinde doğru bir şekilde öğrenmesini ve genelleme yapmasını sağlamak için parametre optimizasyonu sonucuna göre seçilmiştir.

Tablo 3. Model eğitim parametreleri

Şifreleyici (Encoder) Model	Tekrar	Yığın Boyutu	Optimize Edici	Öğrenme Oranı
dbmdz/bert-base-turkish-128k-uncased	4	32	AdamW	5e-5

Tablodaki parametreler kullanılarak 2 farklı veri seti ile 2 farkı model eğitilmiştir. Bu modellerden yoğun veri seti ile eğitilen model “Yoğun Model”, orta yoğun veri seti ile eğitilen model ise “Orta Yoğun Model” olarak adlandırılmıştır.



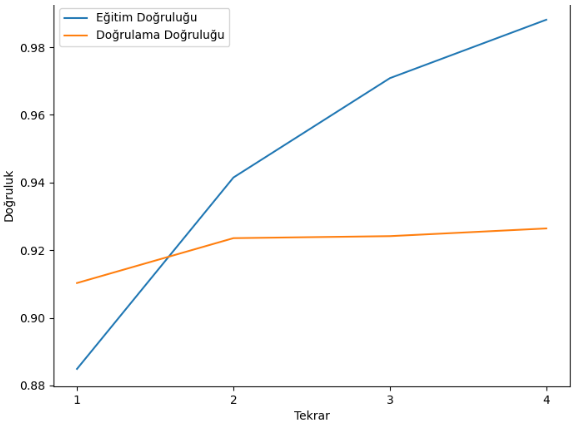
Şekil 8. Yoğun ve orta yoğun model eğitim kaybı

Şekil 8’deki grafikte, "Yoğun Model Kaybı" ve "Orta Yoğun Model Kaybı" olarak iki farklı modelin eğitim kayıplarının dört tekrar boyunca değişimi gösterilmiştir. İlk tekrarda orta yoğun modelin kaybı (yaklaşık 0,5), yoğun modelinkinden (yaklaşık 0,4) daha yüksektir, bu da başlangıçta orta yoğun modelin daha kötü performans gösterdiğini işaret eder. Tekrar sayısı arttıkça her iki modelin de kayıp değerleri azalarak öğrenme sağlanmıştır. Tüm tekrarlar boyunca yoğun modelin kaybı, orta yoğun modelin kaybından daha düşük seyretmiştir, bu da yoğun modelin daha iyi performans sergilediğini gösterir. Son tekrarda ise kayıp değerleri neredeyse eşitlenerek (yaklaşık 0.1) her iki modelin de yeterli eğitim sonrası benzer performans seviyesine ulaştığı görülmüştür.

Tablo 4. Geliştirilen modeller ve skorları

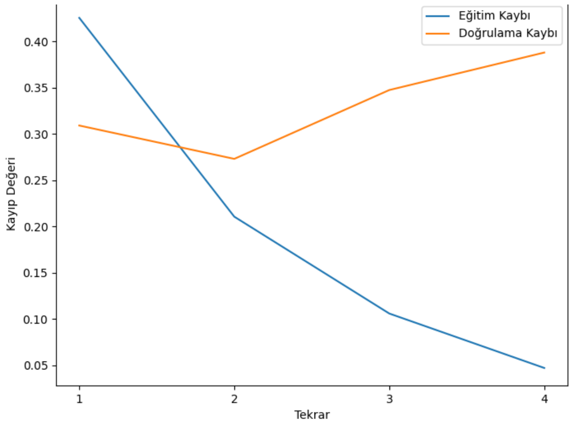
Modeller	F1-Skoru	Kesinlik (Precision)	Duyarlılık(Recall)
Orta Yoğun Model	0,894	0,891	0,897
Yoğun Model	0,912	0,911	0,914

Tablo 4’te geliştirilen iki farklı model arasında yapılan karşılaştırmalar verilmiştir. Yoğun model belirgin bir üstünlük sergilemektedir. Yoğun modelin F1 skoru (0,912) ve kesinlik (0,911) ile duyarlılık (0,914) ölçütleri yüksek ve birbirine yakın değerlerde bulunmaktadır, bu durum modelin hem doğruluğunun hem de hata yapma olasılığının düşük olduğunu göstermektedir. Diğer taraftan, orta yoğun modelin performansı da genel olarak kabul edilebilir düzeydedir (F1 skoru = 0,894, kesinlik = 0,891, duyarlılık = 0,897), ancak yoğun modele kıyasla bir miktar geride kalmıştır. Bu sonuçlar, yoğun modelin özellikle veri dengesizliği gibi zorluklarla daha etkin şekilde başa çıkabildiğini ve sınıflandırma performansının daha istikrarlı olduğunu işaret etmektedir.



Şekil 9. Yoğun model eğitim ve doğrulama doğruluğu

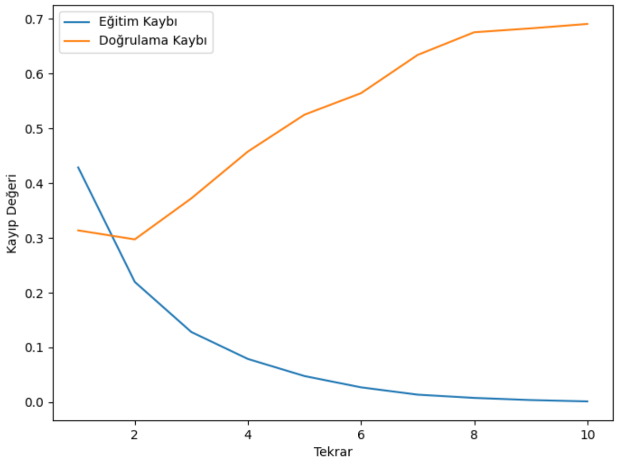
Şekil 9’daki grafikte, yoğun modelin eğitim aşamasındaki doğruluk değerleri görülmektedir. Modelin eğitim doğruluğu her tekrar ile birlikte sürekli artmakta ve dördüncü tekrar sonunda %98’in üzerine çıkmaktadır. Bu, modelin eğitim veri setini oldukça iyi öğrendiğini göstermektedir. Doğrulama doğruluğu ise ilk tekrardan itibaren yavaş bir artış gösterip, ikinci tekrardan sonra yaklaşık %92,5 seviyesinde kalmaktadır. Bunun sebebi ise doğrulama veri setindeki örneklerin eğitim veri setine göre daha çeşitli ve karmaşık olması, modelin doğrulama doğruluğunda daha sınırlı bir artış göstermesine neden olmuştur. Bu durumda, doğrulama doğruluğunun stabil seyretmesi ve aşırı düşüş göstermemesi, modelin genelleme yeteneğinin yeterli olduğunu ve eğitim sürecinde aşırı öğrenme sorunu yaşanmadığını işaret etmiştir.



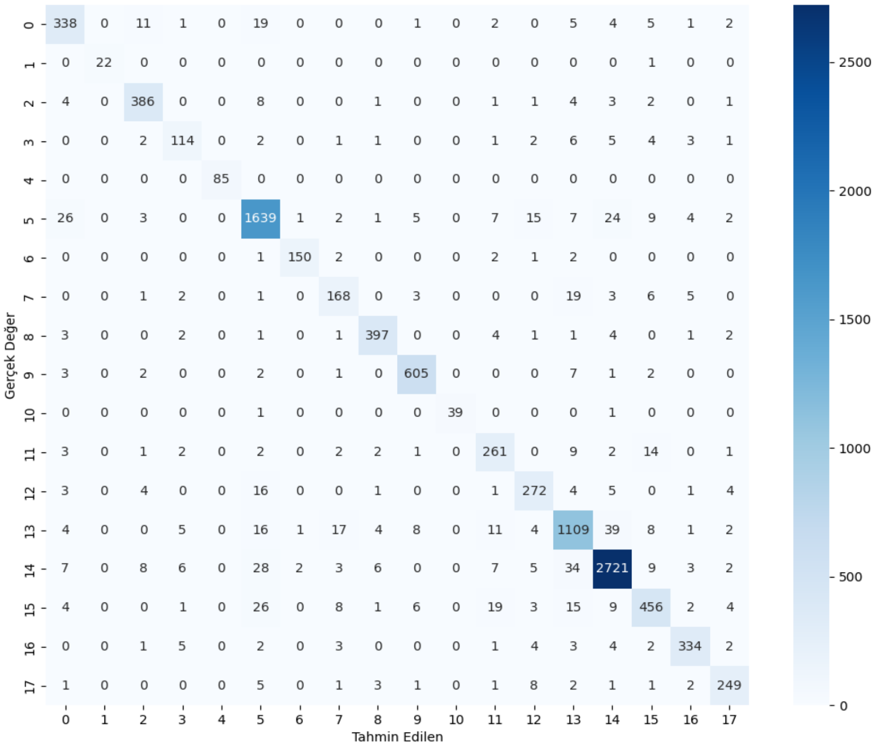
Şekil 10. Yoğun model eğitim ve doğrulama kaybı

Şekil 10'daki grafikte, yoğun modele ait eğitim aşamasındaki kayıp değerleri görülmektedir. Modelin eğitim kaybı her tekrar ile birlikte azalmakta ve dördüncü tekrar sonunda neredeyse sıfıra yaklaşmaktadır. Bu da modelin eğitim veri setini çok iyi öğrendiğini göstermektedir. Doğrulama kaybı ise ilk tekrarda daha düşük başlamakta, ikinci tekrarda bir miktar düşüş göstermekte ancak üçüncü tekrardan itibaren artış eğilimindedir. Doğrulama veri setinde çok çeşitli ve farklı veriler içerdiği düşünüldüğünde, bu artış modelin genelleme yeteneğini zorlayan çeşitliliğe bağlanmaktadır. Bu durum, modelin doğrulama verisindeki hata oranının artmasına neden olmuştur. Ancak, bu çeşitlilik ve doğrulama kaybındaki artışa rağmen, modelin doğrulama doğruluğunun stabil seyretmesi, aşırı öğrenmenin belirgin olmadığını ve modelin geniş veri çeşitliliği karşısında yeterli performans gösterdiğini işaret etmektedir.

Şekil 11'de verilen grafikte 4. tekrardan sonra eğitim kaybı düzenli bir şekilde azalırken doğrulama kaybının belirgin bir şekilde artması, modelin aşırı öğrenme yaşadığını net bir şekilde ortaya koymaktadır. 4. tekrarda doğrulama kaybı, eğitim kaybına kıyasla daha düşükken, takip eden tekrarlar boyunca eğitim kaybı düşmeye devam ederken doğrulama kaybı yükselmiş ve 7. tekrardan itibaren sabitlenmiştir. Bu durum, modelin eğitim verisine aşırı uyum sağladığını ve doğrulama verisinde genelleme yeteneğinin zayıfladığını göstermektedir.



Şekil 11. Yoğun model eğitim ve doğrulama kaybı (10 tekrar



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	338	0	11	1	0	19	0	0	0	1	0	2	0	5	4	5	1	2
1	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	4	0	386	0	0	8	0	0	1	0	0	1	1	4	3	2	0	1
3	0	0	2	114	0	2	0	1	1	0	0	1	2	6	5	4	3	1
4	0	0	0	0	85	0	0	0	0	0	0	0	0	0	0	0	0	0
5	26	0	3	0	0	1639	1	2	1	5	0	7	15	7	24	9	4	2
6	0	0	0	0	0	1	150	2	0	0	0	2	1	2	0	0	0	0
7	0	0	1	2	0	1	0	168	0	3	0	0	0	19	3	6	5	0
8	3	0	0	2	0	1	0	1	397	0	0	4	1	1	4	0	1	2
9	3	0	2	0	0	2	0	1	0	605	0	0	0	7	1	2	0	0
10	0	0	0	0	0	1	0	0	0	0	39	0	0	0	1	0	0	0
11	3	0	1	2	0	2	0	2	2	1	0	261	0	9	2	14	0	1
12	3	0	4	0	0	16	0	0	1	0	0	1	272	4	5	0	1	4
13	4	0	0	5	0	16	1	17	4	8	0	11	4	1109	39	8	1	2
14	7	0	8	6	0	28	2	3	6	0	0	7	5	34	2721	9	3	2
15	4	0	0	1	0	26	0	8	1	6	0	19	3	15	9	456	2	4
16	0	0	1	5	0	2	0	3	0	0	0	1	4	3	4	2	334	2
17	1	0	0	0	0	5	0	1	3	1	0	1	8	2	1	1	2	249

Şekil 12. Karışıklık matrisi

Şekil 12 model performansının karışıklık matrisini göstermektedir. Matrisin satırları gerçek sınıfları, sütunları ise modelin tahmin ettiği sınıfları temsil etmektedir. Diyagonal üzerindeki hücreler doğru tahmin edilen örneklerin sayısını (Doğru Pozitif, TP), diyagonal dışındaki hücreler ise yanlış tahmin edilen örneklerin sayılarını (Yanlış Pozitif, FP ve Yanlış Negatif, FN) gösterir. Örneğin, 0 sınıfının temsil ettiği alışveriş sınıfında doğru tahmin edilen 338 örnek ve 2 sınıfının temsil ettiği ev ve bahçe sınıfında tahmin edilen 11 örnek bulunmaktadır.

5. Tartışma (Discussion)

Bu çalışmada geliştirilen "Yoğun Model" %92,40 doğruluk oranıyla Instagram fenomenlerinin gönderilerini kategorize etmede en yüksek performansı sergilemiştir. Bu başarı, modelin dengeli veri seti üzerinde eğitim alması sayesinde sınıflar arasında adil bir öğrenme gerçekleştirebilmesine ve her bir sınıfı daha doğru tanımlayabilmesine bağlanabilir.

Tablo 5'te verilen sonuçlar, Instagram gönderilerini kullanarak yapılan sınıflandırma çalışmalarının doğruluk oranlarını göstermektedir. Modelin performansı, literatürde yaygın olarak kullanılan sınıflandırma yöntemleri ile karşılaştırıldığında belirgin bir üstünlük göstermektedir. Özellikle GaussianNB, K-En Yakın Komşu, Destek Vektör Sınıflandırması ve Rastgele Orman gibi yöntemlerin doğruluk oranlarının düşük kaldığı göz önüne alındığında, önerilen modelin Instagram gönderilerini sınıflandırmada daha etkili bir yöntem olduğu ortaya çıkmaktadır.

Buna ek olarak, önceki çalışmalarda geliştirilen "Influencer Profiler" modeli %60,90 doğruluk oranı (Kim vd., 2020) ile sınırlı bir başarı elde ederken, bu çalışmada geliştirilen modelin %92,40 doğruluk oranına ulaşması, fenomen-ürün hedef kitle eşleştirilmesinde önemli bir avantaj sunmaktadır. Bu sonuçlar, önerilen modelin sosyal medya pazarlamasında fenomen seçim sürecini daha etkili bir hale getirdiğini ve pazarlama stratejilerinin başarısını artırabileceğini göstermektedir.

Sonuç olarak, önerilen model, diğer çalışmada geliştirilen ve yaygın sınıflandırma algoritmalarından çok daha yüksek bir doğruluk oranı sunarak, Instagram gönderilerini sınıflandırmada etkili bir yöntem olduğunu kanıtlamıştır.

Tablo 5. Instagram gönderilerindeki açıklama metinleri kullanılarak yapılan sınıflandırma çalışmalarının doğruluk oranları (Kim vd., 2020)

Model	Girdi	Doğruluk
GaussianNB	Metin	%40,70
K-En Yakın Komşu	Metin	%38,85
SVC	Metin	%36,20
Rastgele Orman	Metin	%31,80
Influencer Profiler	Metin	%60,90
Orta Yoğun Model	Metin	%90,57
Yoğun Model	Metin	%92,40

Elde edilen performans metriklerine göre 0,92 doğruluk değeri, modelin elde edilen veri setindeki tüm örneklerin %92'sini doğru sınıflandırıldığını belirtir.

Çalışma kapsamında geliştirilen "Yoğun Model" ve "Orta Yoğun Model" arasında "Yoğun Model" en iyi performans gösteren model olmuştur. Yoğun modele ait diğer performans metrikleri Tablo 6'da verilmiştir.

Tablo 6. Model performans değerlendirme metrikleri

Performans Metriği	Hesaplanan Değeri
Doğruluk	0,92
Kesinlik	0,91
Duyarlılık	0,91
F1 Skoru	0,91

Kesinlik ve duyarlılık değerlerinin 0,91 olması, modelin pozitif sınıflandırmalarda güvenilir olduğunu göstermektedir. F1 skoruna bakıldığında ise modelin hem kesinlik hem duyarlılık açısından güçlü olduğunu göstermektedir. Genel olarak model performans metrikleri 0,91'in üzerinde olduğundan sınıflandırma model performansının yüksek olduğu söylenebilir.

Kesinlik ve duyarlılık metriklerinin birbirine yakın ve yüksek olması, modelin pozitif sınıfları doğru bir şekilde tanımladığını ve çok fazla yanlış pozitif üretmediğini göstermektedir. Bu, modelin etkinliğinin dengeli olduğunu ve herhangi bir ölçümde belirgin bir zayıflığın bulunmadığını göstermektedir. Çalışma sonunda elde edilen sınıflandırma modelinin gerçek dünya uygulamalarında güvenilir bir şekilde kullanılabileceği ve tutarlı sonuçlar vereceği yargısı elde edilmiştir.

6. Sonuçlar (Conclusions)

Gerçekleştirilen çalışma ile, Instagram fenomenlerinin gönderi açıklamalarını analiz ederek otomatik bir şekilde fenomenlerin hitap ettiği hedef kitleyi tespit edebilen dönüştürücü mimarisi tabanlı bir sınıflandırma modeli geliştirilmiştir. Türkçe dilinde oluşturulan veri seti üzerinde eğitilen sınıflandırma model performansı değerlendirildiğinde 0,92 doğruluk ve 0,91 F1 skoru değeri elde edilmiştir. Model performans değerlendirmesi sonucunda, gerçek dünya uygulamalarında başarılı ve güvenilir bir şekilde kullanılabilecek bir sınıflandırma modeli elde edildiği söylenebilir.

Bu çalışma, uygulama alanı değerlendirildiğinde literatüre özgün bir katkı sağlamıştır. Gelecek çalışmalarda, gönderilere ait resimler kullanılarak bir resim sınıflandırma modeli geliştirilecektir. Bu model sayesinde, gönderi resimlerinin ait olduğu kategoriler tespit edilebilecektir. İleriki çalışmalarda, daha büyük bir veri seti elde edilerek model performansı büyük hacimli veri üzerinde değerlendirilecek ve gerekli optimizasyonlar yapılacaktır. Ayrıca, veri setinin büyütülmesi ile birlikte kategori sayısında da artış sağlanacak ve böylelikle daha geniş bir alanda veri sınıflandırılması mümkün olacaktır. Bu kapsamda, resim ve metin verilerinin bir arada kullanımı, modelin doğruluğunu ve genelleme kabiliyetini artıracak, araştırma alanına önemli katkılar sağlayacaktır.

7. Teşekkür (Acknowledgment)

Bu çalışma Küçük ve Orta Ölçekli İşletmeleri Geliştirme ve Destekleme İdaresi Başkanlığı (KOSGEB) tarafından Ar-Ge, Ür-Ge ve İnovasyon projesi kapsamında desteklenmiştir.

Kaynaklar (References)

Acı, Ç. and Çırak, A., 2019. Türkçe haber metinlerinin konvolüsyonel sinir ağları ve Word2Vec kullanılarak sınıflandırılması. Bilişim Teknolojileri Dergisi, 12(3), pp.219-228.

Aitken, K., Ramasesh, V., Cao, Y. and Maheswaranathan, N., 2021. Understanding how encoder-decoder architectures attend. Advances in Neural Information Processing Systems, 34, pp.22184-22195.

Apify. (2022). Web scraping, data extraction and automation. Apify. Retrieved March 22, 2022, from: https://apify.com/

Arslan, E., 2017, August. The effect of social media on marketing. In International Congress Of Eurasian Social Sciences (ICOESS).

Bagwell, K., 2007. The economic analysis of advertising. Handbook of industrial organization, 3, pp.1701-1844.

Carr, C.T. and Hayes, R.A., 2015. Social media: Defining, developing, and divining. Atlantic journal of communication, 23(1), pp.46-65.

Çelıkten, A. and Bulut, H., 2021, June. Turkish medical text classification using bert. In 2021 29th signal processing and communications applications conference (SIU) (pp. 1-4). IEEE.

Çepni, S., Toprak, A. G., Yatkınoğlu, A., Mercan, Ö. B., & Ozan, Ş. (2023). Performance Evaluation of a Pretrained BERT Model for Automatic Text Classification. Journal of Artificial Intelligence and Data Science, 3(1), 27-35.

Çopuroğlu, F., 2022. Fenomen pazarlamanın satın alma niyeti üzerindeki etkisinde menşei ülkenin aracılık rolü. Gaziantep University Journal of Social Sciences, 21(4), pp.2258-2275.

Dalkılıç, G., & Çebi, Y., 2003. Türkçe külliyat oluşturulması ve Türkçe metinlerde kullanılan kelimelerin uzunluk dağılımlarının belirlenmesi. Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi, 5(1), 1-7.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

González-Carvajal, S. and Garrido-Merchán, E.C., 2020. Comparing BERT against traditional machine learning text classification. arXiv preprint arXiv:2005.13012.

Grootendorst, M., 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv preprint arXiv:2203.05794.

Güler, H., Şahinkayası, Y. and Şahinkayası, H., 2017. İnternet ve mobil teknolojilerin yaygınlaşması: Fırsatlar ve sınırlılıklar. Kilis 7 Aralık Üniversitesi Sosyal Bilimler Dergisi, 7(14), pp.186-207.

Karataş, M. and Eti, H.S., 2022. Dijital pazarlama çağında Instagram fenomenlerinin tüketici satın alma davranışlarına etkisi. AJIT-e: Academic Journal of Information Technology, 13(50), pp.184-219.

Kaşıkçı, T. and Gökçen, H., 2014. Metin madenciliği ile e-ticaret sitelerinin belirlenmesi. Bilişim Teknolojileri Dergisi, 7(1).

Khalid, U., Beg, M.O. and Arshad, M.U., 2021. Rubert: A bilingual roman urdu bert using cross lingual transfer learning. arXiv preprint arXiv:2102.11278.

Kim, S., Jiang, J.Y., Nakada, M., Han, J. and Wang, W., 2020, April. Multimodal post attentive profiling for influencer marketing. In Proceedings of The Web Conference 2020 (pp. 2878-2884).

Leung, F.F., Gu, F.F. and Palmatier, R.W., 2022. Online influencer marketing. Journal of the Academy of Marketing Science, 50(2), pp.226-251.

Liu, C.Z., Sheng, Y.X., Wei, Z.Q. and Yang, Y.Q., 2018, August. Research of text classification based on improved TF-IDF algorithm. In 2018 IEEE international conference of intelligent robotic and control engineering (IRCE) (pp. 218-222). IEEE.

Özdemir, S.S., Özdemir, M., Polat, E. and Aksoy, R., 2014. Sosyal medya kavrami ve sosyal ağ sitelerinde yer alan online reklam uygulamalarinin incelenmesi. Ejovoc (Electronic Journal of Vocational Colleges), 4(4), pp.58-64.

Özkan, M. and Kar, G., 2022. Türkçe Dilinde Yazılan Bilimsel Metinlerin Derin Öğrenme Tekniği Uygulanarak Çoklu Sınıflandırılması. Mühendislik Bilimleri ve Tasarım Dergisi, 10(2), pp.504-519.

Öztek, M., Yerden, N.K., Çolak, E. and Sarı, E., 2021. Fenomen pazarlamasında sosyal medyanın rolü ve moda sektörü üzerine bir içerik analizi. Yaşar Üniversitesi E-Dergisi, 16(62), pp.1053-1077.

Prabowo, F. and Purwarianti, A., 2017, November. Instagram online shop's comment classification using statistical approach. In 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE) (pp. 282-287). IEEE.

Reimers, N. and Gurevych, I., 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

Sel, S. and Hanbay, D., 2019, April. E-mail classification using natural language processing. In 2019 27th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

Terkan, R., 2014. Sosyal Medya Ve Pazarlama: Tüketicide Kalite Yansimasi. Organizasyon ve Yönetim Bilimleri Dergisi, 6(1), pp.57-71.

Uslu, O. and Özmen-akyol, S., 2021. Türkçe haber metinlerinin makine öğrenmesi yöntemleri kullanılarak sınıflandırılması. Eskişehir Türk Dünyası Uygulama ve Araştırma Merkezi Bilişim Dergisi, 2(1), pp.15-20.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017.

Attention is all you need. Advances in neural information processing systems, 30.

Yıldırım, Y. and Yıldırım, H., 2022. Dijital Sınırların Sonsuzluğu: Günlük Hayattan Somut Örnekler. Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 10(4), pp.1838-1864.