Middle East Journal of Science

https://dergipark.org.tr/mejs

MEJS

e-ISSN:2618-6136

Research Article

COMPARATIVE FEATURE SELECTION APPROACHES FOR ALZHEIMER'S DISEASE USING GENETIC ALGORITHMS AND PARTICLE SWARM OPTIMIZATION

Simay TÜRKÜSAY [1], *Murat OTURAKÇI** [1], *Esra EKİNCİ* [1], *Deniz TÜRSEL ELİİYİ* [1]

[1]Department of Industrial Engineering, Faculty of Engineering and Architecture, İzmir Bakırçay University, İzmir, Türkiye
*Corresponding author; murat.oturakci@bakircay.edu.tr

Abstract: *Alzheimer's disease (AD) is the most prevalent form of dementia, significantly impairing cognitive abilities such as memory and judgment. The number of dementia cases is expected to rise dramatically in the coming decades, with Alzheimer's disease accounting for 60-80% of these cases. Early detection is crucial for improving patient outcomes, yet diagnosing Alzheimer's at its early stages remains challenging due to various clinical and perceptual obstacles. This study addresses whether Alzheimer's can be detected in advance and the methods that can be used for early diagnosis. Using an Alzheimer's disease dataset sourced from Kaggle including 2,150 samples with 32 independent and 1 dependent variables, various classification algorithms were applied to assess performance. Feature selection techniques, including both classical and metaheuristic methods (Genetic Algorithm and Particle Swarm Optimization), were then applied to the dataset. These methods helped reduce the dataset's dimensionality while maintaining high diagnostic performance. The results showed that both metaheuristic algorithms selected 14 variables, producing the same high performance rate of 95.57% compared to the initial 32 variables. The findings suggest that Alzheimer's disease can be detected more efficiently with fewer variables, reducing analysis time and increasing diagnostic speed. Metaheuristic algorithms, particularly Particle Swarm Optimization, proved to be the most effective, enhancing the performance of 33 classifiers, while the Genetic Algorithm improved the performance of 28 classifiers. This study demonstrates that Alzheimer's can be detected with fewer variables, in less time, and with a higher accuracy rate. As a result, improved patient outcomes through reduced computational complexity and enhanced diagnostic efficiency can potentially be achieved.*

Keywords: *Alzheimer's Disease; Classification, Feature Selection, Genetic Algorithm, Particle Swarm Optimization*

Received:December 23, 2024 *Accepted: April 21, 2025*

1. Introduction

Dementia is a general medical term for a decline in cognitive function severe enough to interfere with daily activities. Alzheimer's disease (AD) is the most common form of dementia, which occurs at least two-thirds of dementia cases that are aged 65 and over [1]. A neurodegenerative condition, Memory, comprehension, language, attention, thinking, and judgment are among the behavioral and cognitive skills that are gradually compromised by Alzheimer's disease [1].

The World Health Organization (WHO) projects that the number of people with dementia, which is already over 55 million, will increase to 75 million by 2030 and 132 million by 2050[2]. According to studies, 60–80% of dementia cases are attributable to Alzheimer's disease, and a new case is diagnosed every three seconds. In Turkey, 5.5% of people 65 and older have Alzheimer's disease, according to the

2022 Turkey Health Survey Report released by the Turkish Statistical Institute (TÜİK). This emphasizes how urgently Turkey, like the rest of the world, must implement effective measures to battle Alzheimer's disease [2].

Clinicians have now been urged to detect Alzheimer's sooner before individuals have developed dementia from the condition. The ability of clinicians to accurately and early detect the underlying pathology and symptoms of Alzheimer's disease is critical to the screening, diagnosis, and treatment of such individuals. Additionally, it helps patients, and their caregivers create future plans and modify their lifestyles in ways that may help them live longer and maintain a higher quality of life. Unfortunately, several challenges can make it challenging to detect early-stage Alzheimer's disease in clinical practice. These include clinicians' time constraints, the challenge of accurately diagnosing Alzheimer's pathology, and patients' and healthcare professionals' propensity to dismiss symptoms as a normal aspect of aging. [3].

This situation prompts two essential questions:

1. Is it possible to reveal the presence or onset of Alzheimer's disease before symptoms appear?
2. If early detection is feasible, what methods or techniques can be employed to identify the disease?

Motivated by these research questions, this study aims to address them by utilizing a previously measured dataset, applying classification algorithms, and implementing feature selection techniques through classical and metaheuristic methods. The main goal is to facilitate the early diagnosis of the disease and enable the immediate initiation of treatment methods.

In this context, various classification algorithms were applied to a pre-measured dataset, and their performance was evaluated. Subsequently, feature selection was performed on the same dataset using classical and metaheuristic algorithms, specifically genetic algorithm and particle swarm optimization. These algorithms are selected as ideal ones for effectively reducing the dimensionality of medical datasets and eliminating overfitting and computational inefficiency [4]. By optimizing the selection of the most informative variables through these algorithms, the most critical predictors of Alzheimer's disease can be utilized for an accurate and efficient medical diagnosis. Based on the results of each algorithm, datasets tailored to the selected variables were created, and classification algorithms were applied to these datasets again. The performance results obtained were then compared with the initial ones. As a result of the comparison, the classifier and feature selection method that yielded the highest performance were identified. The study concluded that the disease identification process could achieve higher performance with fewer variables than those initially examined in the dataset.

This study provides both theoretical and practical contributions to Alzheimer’s disease. Theoretically, it highlights the effectiveness of combining feature selection with machine learning techniques, demonstrating that fewer but more relevant features can significantly improve classification performance. Practically, the research shows that using metaheuristic algorithms like genetic algorithms and particle swarm optimization for feature selection can enhance diagnostic efficiency, reducing computational complexity while maintaining high accuracy. These findings offer valuable insights for developing more efficient early and cost-effective Alzheimer’s disease in clinical settings. The diagnostic process for this critical disease can possibly be streamlined through faster and more reliable predictions as a result.

The study is organized into several key sections, each addressing a specific aspect of the research process. It begins with a Literature Review, where previous studies and relevant research are examined to provide context and highlight existing gaps in the field. This is followed by the Methods and Materials section, which outlines the research design, data collection techniques, and tools used in the analysis. In the Results section, the findings of the study are presented, including any statistical analyses or observations made during the investigation. Finally, the study concludes with a Conclusion section, summarizing the key insights, discussing the implications of the findings, and suggesting areas for future research.

2. Literature Review

Within the scope of the literature review, notable studies using feature selection for the detection of Alzheimer's disease between 2014 and 2024 are presented. This study outlines the methods utilized and provides an overview of various studies conducted between 2014 and 2024 that demonstrate the applications of these methods.

Anirudha et al. (2014) present a Genetic Algorithm-based Wrapper feature selection Hybrid Prediction Model (GWHPM) for disease prediction, utilizing k-means clustering to remove outliers and a genetic algorithm for optimal feature selection. These features are then used to build classifier models including Decision Tree, Naïve Bayes, k-nearest neighbor, and Support Vector Machine, with comparative results showing that the proposed GWHPM outperforms existing methods. Mirzaei et al. (2018) aim to develop a non-invasive method for early detection of Alzheimer’s disease using voice analysis techniques combined with machine learning algorithms. Neelaveni et al. (2020) apply machine learning algorithms using psychological parameters, including age, number of visits, MMSE scores, and education level, to predict Alzheimer’s disease, highlighting the importance of early prediction in mitigating disease progression, despite AD typically being diagnosed in later stages. Saputra et al. (2020) classify Alzheimer’s disease using different decision tree algorithms with feature selection through the Particle Swarm Optimization (PSO) algorithm, achieving a 93.56% accuracy after applying PSO-based feature selection to the OASIS 2 dataset. Ramaswamy et al. (2021) identify genes contributing to Alzheimer’s disease using gene expression data from the human brain in AD patients and older control subjects, applying a two-step gene selection method that combines statistical techniques and heuristic optimization, with classifiers achieving 100% accuracy on the GSE5281 test dataset. Divya et al. (2021) investigate the classification of Alzheimer’s disease, mild cognitive impairment (MCI), and normal control (NC) using MRI images from the ADNI dataset, finding that dimensionality reduction improves classification performance, especially with limited high-dimensional data, and achieving accuracy rates of 96.82%, 89.39%, and 90.40% for NC/AD, NC/MCI, and MCI/AD classifications, respectively. Buyrukoğlu (2021) develops a model for early Alzheimer’s disease diagnosis using ensemble feature selection methods, with Random Forest achieving the highest performance at 91% across three target classes: Normal (CN), MCI, and AD. Noroozi et al. (2023) analyze feature selection techniques for heart disease prediction using the Cleveland Heart Disease dataset, showing that feature selection enhances some algorithms (e.g., J48) but diminishes others (e.g., MLP, RF), with the SVM-based filtering method achieving the highest accuracy at 85.5%. Hassouneh et al. (2024) investigate the significance of fused texture features derived from 3D MRI and PET images for the early detection of Alzheimer’s disease, finding that GLCM texture features from the hippocampus and entorhinal cortex outperform volume and SUVR features, achieving 90% sensitivity in identifying MCI converters with minimal false positives, and highlighting the role of various feature types in improving classification accuracy for early AD diagnosis.

3. Materials and Methods

3.1. Materials

This study examines an anonymous Alzheimer's disease dataset sourced from the Kaggle platform, which contains high-quality data [13]. The dataset used consists of 33 variables (32 independent and 1 dependent) and 2,150 samples. The independent variables include: “Age, Gender, Ethnicity, Education Level (EL), BMI, Smoking, Alcohol Consumption (AC), Physical Activity (PA), Diet Quality (DQ), Sleep Quality (SQ), Family History Alzheimers (FHA), Cardiovascular Disease (CD), Diabetes, Depression, Head Injury (HI), Hypertension (HT), SystolicBP (SBP), DiastolicBP(DBP), Cholesterol Total (CT), Cholesterol LDL (LDL), Cholesterol HDL (HDL), Cholesterol Triglycerides (TG), MMSE, Functional Assessment (FA), Memory Complaints (MC), Behavioral Problems (BP), ADL, Confusion (CO), Disorientation (DO), Personality Changes (PC), Difficulty Completing Tasks (DCT), Forgetfulness (FF)", while the dependent variable is "Diagnosis." To ensure accurate measurements and proper numerical interpretation by the algorithms, the values of certain variables, initially presented in numerical categories, were modified following the guidelines provided on the dataset's hosting platform. Specifically: For Gender, values were encoded as 0 for Male and 1 for Female. For Ethnicity, values were encoded as 0 for Caucasian, 1 for African American, 2 for Asian, and 3 for Other. For Education Level, values were encoded as 0 for None, 1 for High School, 2 for Bachelor's, and 3 for Higher. Other categorical variables with binary values were encoded as 0 for No and 1 for Yes. A part of the dataset is presented in Table 1 as an example. While analyzing the model, the data was separated into test (30%) and training (70%). This separation was performed randomly to ensure that both sets of evolutions represented the general distribution.

Table 1. Dataset Sample [13]

Age	**Gender**	**Ethnicity**	**EL**	**BMI**	**Smoking**	**AC**	**PA**	**DQ**	**SQ**	**FH**
73	0	0	2	22,928	0	13,297	6,327	1,347	9,026	0
89	0	0	0	26,828	0	4,543	7,620	0,519	7,151	0
73	0	3	1	17,796	0	19,555	7,845	1,826	9,674	1
74	1	0	1	33,801	1	12,209	8,428	7,436	8,393	0
89	0	0	0	20,717	0	18,454	6,310	0,795	5,597	0
86	1	2	1	30,627	0	4,140	0,211	1,585	7,262	0
68	0	3	2	38,388	1	0,646	9,258	5,897	5,478	0
75	0	0	1	18,776	0	13,724	4,649	8,342	4,213	0
72	1	1	0	27,833	0	12,168	1,531	6,737	5,748	0
87	0	0	0	35,456	1	16,029	6,441	8,086	7,552	0
CD	**Diabetes**	**Depression**	**HI**	**HT**	**SBP**	**DBP**	**CT**	**LDL**	**HDL**	**TG**
0	1	1	0	0	142	72	242,367	56,151	33,683	162,189
0	0	0	0	0	115	64	231,163	193,408	79,028	294,631
0	0	0	0	0	99	116	284,182	153,323	69,772	83,638
0	0	0	0	0	118	115	159,582	65,367	68,457	277,577
0	0	0	0	0	94	117	237,602	92,870	56,874	291,199
0	1	0	0	0	168	62	280,713	198,335	79,081	263,944
0	0	0	1	0	143	88	263,734	52,471	66,533	216,489
0	0	0	0	0	117	63	151,383	69,624	77,347	210,571
0	0	0	0	1	117	119	233,606	144,046	43,076	151,164
1	0	0	0	0	130	78	281,630	130,498	74,291	144,176
MMSE	**FA**	**MC**	**BP**	**ADL**	**CO**	**DO**	**PC**	**DCT**	**FF**	**Diagnosis**
21,464	6,519	0	0	1,726	0	0	0	1	0	0
20,613	7,119	0	0	2,592	0	0	0	0	1	0
7,356	5,895	0	0	7,120	0	1	0	1	0	0
13,991	8,965	0	1	6,481	0	0	0	0	0	0
13,518	6,045	0	0	0,015	0	0	1	1	0	0
27,518	5,510	0	0	9,016	1	0	0	0	0	0
1,964	6,062	0	0	9,236	0	0	0	0	1	0
10,140	3,401	0	0	4,517	1	0	0	0	1	1
25,821	7,396	0	1	0,756	0	0	1	0	0	0
28,388	1,149	0	1	4,554	0	0	0	0	0	0

3.2. Methods

Classification is known as one of the most widely used data mining methods for predicting group memberships. Classification techniques work on the basis of assigning data points to predefined classes or groups. Many classification methods have different strengths depending on the characteristics of the data set [14]. In addition to basic methods such as decision trees and support vector machines, there are also classification methods with complex structures such as neural networks. In this study, 45 classification algorithms were used according to the suitability of the selected data set structure and the diversity of the algorithms allowed for a comparative evaluation of their performances.

BayesNet: The Bayes theorem is the foundation of BayesNet. Thus, a Bayesian network is created by calculating the conditional probability of each node in BayesNet. A directed acyclic graph is the Bayesian Network. All attributes are assumed to be nominal in BayesNet, and any missing values are replaced globally [15].

NaiveBayes: Because of its robustness, elegance, and simplicity, NaiveBayes is frequently employed for categorization. Navie and Bayes are two ways to describe NavieBayes. When the occurrences are independent and the Bayes rule is applied, Navie, which stands for independence, is true to multiply probability. This method assumes that a class's characteristics are independent in practice. When the data set is real, the NavieBayes perform better [15].

NaiveBayesMultinomialTest: One of the most popular text mining techniques is the Naïve Bayes classifier. Multinomial Naïve Bayes, which is essentially an improved version of the original Naïve Bayes classifier, efficiently manipulates the word count by determining the frequency of each word, whereas in the Naïve Bayes classifier, the frequency of the words has little bearing on how the algorithm operates. It is well recognized that a text's frequency has a greater influence on its classification into several groups. Therefore, Multinomial Naïve Bayes is thought to be the best method for text categorization [16].

NaiveBayesUpdateable: This is the NaiveBayes version that can be updated. When buildClassifier, often referred to as incremental update, is invoked with zero training examples, this classifier will utilize a default precision of 0.1 for numeric attributes [15].

LibSVM: A programming library for SVM is called LibSVM. Researchers utilize it for tasks involving regression and classification. WEKA, which includes a set of machine learning algorithms for data mining, also incorporates LibSVM [17]. ***Logistic:*** Regarding binary classification A lot of people utilize logistic regression. A linear and additive summary of a variable's impact on the logged chances of possessing a characteristic on an event is given by the logistic regression coefficients. In this case, the result is determined by one or more independent variables. There are two possible outcomes: 1 for true and 0 for false. It is employed to shed light on data and explain how one dependent binary variable and one or more independent variables relate to one another [18].

MultiLayerPerceptron: Neural networks and artificial intelligence are unqualified definitions of multilayer perceptrons. A feedforward neural network with one or more layers between the input and output layers is called a multilayer perceptron (MLP) [15].

SGD: An effective technique known as incremental gradient descent is stochastic gradient descent or SGD. Additionally, it is regarded as a stochastic approximation method that proceeds in this direction while decreasing exponentially by averaging previous gradients. Consequently, it is an optimization method with a number of advantages; for instance, it offers both optimal runtime and ideal sample complexity needs [19].

SGDText: To determine the model parameters that best fit the expected and actual outputs, machine learning applications frequently employ the optimization process known as stochastic gradient descent. It's a strong yet imprecise method. This model disregards non-string (text) inputs, which is how it differs from the conventional SGD classifier [20].

SimpleLogistic: The goal of the well-liked statistical analysis method SimpleLogistic is to identify the best linear logistic regression model. With basic regression functions, it is comparable to the LogitBoost approach. This algorithm, which relies on the logistic function, simulates the outcome's log odds rather than the actual result. Additionally, SimpleLogistic describes the connection between one or more independent variables and the category dependent variable [19].

SMO: Sequential Minimal Optimization, or SMO, is an improved technique for SVM training that has demonstrated strong performance across a variety of issues. However, due to its implementation challenges and training complexity, SVM's employment was constrained. Because SMO is conceptually straightforward, simple to implement, and generally faster than SVM, it is thus subtly enhanced [19].

VotedPerception: Voted perceptron methods for analyzing tiny samples and exploiting the most significant margin of data [15].

IBK: IBK, or instance-based k-nearest neighbors, is a straightforward technique that expands on the k-nearest neighbors algorithm by lowering the amount of storage space needed. To effectively categorize target points (unknown class) based on their distances from reference points that make up a training sample in which their class is already known, IBK uses similarity computations between instances, similar to those of KNN [19].

KStar: K-star, sometimes known as K*, is a classifier that is instance-based. A similarity function determines the class of a test instance based on the training examples that are similar to it. It employs an entropy-based distance function, which sets it apart from other instance-based learners. Instance-based learners use a database of previously categorized examples to classify an instance [21].

LWL (Locally Weighted Learning): One of the key algorithms in lazy learning is locally weighted learning (LWL). When a new instance needs to be processed, a weighted set of training instances linked to the test instance is determined by calculating the distance between the training and test instances using a distance function. This weighted set is then used to build a new model to process the new instance. In conventional weighted learning methods, the Euclidean distance is typically utilized to calculate the separation between instances [22].

IterativeClassifierOptimizer: Through the use of cross-validation, the ICO method was developed to maximize the number of iterations in each performance. Numerical, nominal, binary, and empty nominal characteristics are among the missing, nominal, and binary classes that this algorithm can handle with ease [23].

AdaBoostM1: Boosting is the process of making any algorithm perform better. Boosting is mostly used to lessen a poor algorithm's flaws. A powerful classifier is built using this algorithm. It was created to enhance the performance of crucial activities by combining numerous different algorithms [24].

AttributeSelectedClassifier: Before being sent to a classifier, attribute selection lowers the dimensionality of the training and test data [25].

Bagging: One of the first and most straightforward integration techniques with the best results was the Bagging algorithm, which Breiman proposed. The fundamental idea is to employ an original training set and a weak classification algorithm. Classifiers are trained using the learning method over several rounds [22].

ClassificationViaRegression: Regression-based classification is taught in this class. One regression model is constructed for each class value once the classes are binarized [26].

CvParameterSelection: Class for selecting parameters for any classifier using cross-validation [27].

FilteredClassifier: This class is used to run an arbitrary classifier on data that has been filtered arbitrarily. Similar to the classifier, the filter's structure is only determined by the training data, and it processes test examples without altering their structure. The instances and/or attributes are resampled with replacement based on the weights before being provided to the filter or the classifier (as appropriate) if there are unequal instance weights or attribute weights and the filter or the classifier cannot handle them [28].

LogitBoost: A course for carrying out logistic regression with addition. This class can handle multi-class problems and uses a regression strategy as the basic learner for classification [29].

MultiClassClassifier: A metaclassifier that uses 2-class classifiers to handle multi-class datasets. For improved accuracy, this classifier can additionally apply output codes that fix errors. Before being sent to the base classifier, the data will be resampled with replacement based on the weights if the instance weights are not uniform and the base classifier is unable to handle them [30].

MultiClassClassifierUpdateable: A metaclassifier that uses 2-class classifiers to handle multi-class datasets. For improved accuracy, this classifier can additionally apply output codes that fix errors [31].

MultiScheme: Class for choosing a classifier from a number of them based on performance on the training data or cross validation on the training data. Performance is evaluated using mean-squared error (regression) or percent correct (classification) [32].

RandomCommittee: Using distinct random number seed values, a random committee builds several base classifiers. The average of the predictions produced by each base classifier is used to determine the final classification outcome [33].

RandomizableFilteredClassifier: Using data that has been subjected to an arbitrary filter, this technique applied an arbitrary classifier. Like the classifier, the filter's structure only functioned with the training data; test instances are processed by the filter without having their structure changed [34].

RandomSubspace: With feature space subsampling, this method aims to give students diversity. The same training data is used to build each component model, but each one adds diversity to the ensemble by considering a randomly selected subset of features. Generally speaking, the number of features is set at the same level for every committee component. An ensemble uses either weight voting or majority voting to decide on classification [35].

Stacking: Historically, one of the earliest techniques for ensemble learning was stacking. By using a "meta-learner" (high-level model) that accepts the output values of the base models as inputs, it integrates multiple base models (lower-level models) constructed using completely distinct classes of machine learning techniques [36]. ***Vote:*** Class for classifier combination. There are various combinations of categorization probability estimations available [37].

WeightedInstancesHandlerWrapper: The primary benefit of the WIHW model is that training instances are weighted using a wrapper technique. The WIHW algorithm employs the resampling with weights technique if the basis classifier is not implementing the core. By default, the training dataset is transferred to the base classifier, and it has the ability to control instance weights [38].

InputMappedClassifier: A wrapper classifier that resolves conflicting training and test data by creating a mapping between the structure of the incoming test instances and the training data used to generate the classifier. Both incoming nominal attribute values that the classifier has never seen before and model attributes that are absent from the incoming examples are given missing values. It is possible to load an existing classifier from a file or train a new one [39].

DecisionTable: A significant component of the categorization process is the use of algorithms based on decision trees. Their primary benefits are that they are typically very quick to compute and do not rely on assumptions about the distribution of data [40].

JRip: William W. Cohen suggested Repeated Incremental Pruning to Produce Error Reduction (RIPPER) as an improved version of IREP, and this class implements this propositional rule learner [41].

OneR: OneR, which stands for "One Rule," is a straightforward but precise classification algorithm that choose the rule with the lowest overall error as its "one rule" after producing one rule for each predictor in the data. We must build a frequency table for every predictor against the target before we can develop a rule for it [42].

PART: In 1998, Eibe Frank and Ian H. Witten developed the partition and regression tree (PART) decision tree algorithm, which employs incomplete decision trees to extract rules from a dataset [22].

ZeroR: Most classes in training data are predicted by the ZeroR classifier. It forecasts the mode for nominal class and the mean for numerical values [43].

DecisionStump: One-level decision trees make up a Decision Stump, a type of machine learning model. That is, it is a decision tree with a single internal node (the root) that is directly linked to the leaves, which are the terminal nodes. Only one input feature's value is used by a decision stump to produce a prediction [44].

HoeffdingTree: An method called the Hoeffding Tree is used to create decision trees from data that increases gradually. The Hoeffding Tree implies that the distribution of data does not change over time and operates with massive streams of data. The Hoeffding bound, which calculates the number of samples required to evaluate some valuable information within a required correctness, is the foundation of the Hoeffding Tree [45].

J48: The decision tree is J48's output. The root, intermediate, and leaf nodes of a tree structure are comparable to those of a decision tree. Every node in the tree has a decision, and our outcome is the effect of that decision. A data set's input space is divided into mutually exclusive sections by decision trees, and each section's data points are described by a label, value, or action. The optimum attribute to split the training data section tree that reaches a specific node is determined using the splitting criterion [15].

LMT (Logistic Model Trees): A decision tree induction and logistic regression (LR) model were combined to create the new supervised learning model known as LMT (logistic model tree), which combines the two classification algorithms to benefit from both approaches [19].

RandomForest: A collaborative learning system for regression, classification, and other tasks, random decision forests are run by formatively generating a large number of decision trees during training. For decision trees that are overly appropriate for their instruction set, random decision forests are suitable. The stated training error is an unbiased estimator cross-validated error rate, which is one of Random Forest's key characteristics [18].

RandomTree: A decision tree based on a random subset of qualities is called a random tree. A collection of nodes and their branches is called a decision tree. A decision tree's node denotes an attribute test, and each branch shows the result of that test. A decision tree's leaves represent the ultimate choice made after calculating each attribute as class labels. A classification rule is formed by the route taken from a root to a leaf [33].

RepTree: A quick decision tree learner is the REP Tree (reduced error pruning tree) algorithm. Information gain and variance are used to construct a decision/regression tree, which is then pruned using reduced-error pruning (with back-fitting). For numeric attributes, the method sorts the data only once. The related instances are divided into pieces in order to handle missing values [46].

One of the most common and significant methods in data preprocessing is feature selection, which is now an essential part of the machine learning procedure. In statistics and machine learning, it is sometimes referred to as variable selection, attribute selection, or variable subset selection. It involves finding pertinent traits and eliminating data that is superfluous, redundant, or noisy. This procedure promotes comprehensibility, boosts predictive accuracy, and expedites data mining techniques [47].

As part of the study, feature selection was performed using three types of algorithms. The first was the CfsSubsetEval algorithm, while the other two were meta-heuristic algorithms: the Genetic Algorithm and the Particle Swarm Optimization Algorithm.

CfsSubsetEval: It assesses the value of a subset of features by taking into account each feature's predictive power on its own as well as the level of overlap between them [48]. ***Genetic Algorithms:*** Natural selection is the focus of optimization techniques known as genetic algorithms. GAs were first proposed by John Holland, who used them to explain how natural systems adapt and to create new artificial structures based on the same ideas. This starts with artificial individuals (represented by a "chromosome" population) and mimics the process of natural selection. GA uses genetic operators (e.g., crossover and mutation) to try to enhance the fitters. Furthermore, it aims to develop chromosomes that are stronger than those of their parents in a specific quantitative metric. As a result, GA has been a popular method for data mining feature selection in recent years [49]. ***Particle Swarm Optimization:*** Kennedy and Eberhart introduced PSO, an evolutionary computation method, in 1995. Fish schools and flocks of birds are examples of social behaviors that drive PSO. The fundamental idea behind PSO is that social contact maximizes knowledge in a population where social and personal thinking coexist. The foundation of PSO is the idea that every solution can be thought of as a particle in a swarm. [50]

The workflow of the study, based on the provided information, is shown in Figure 1.

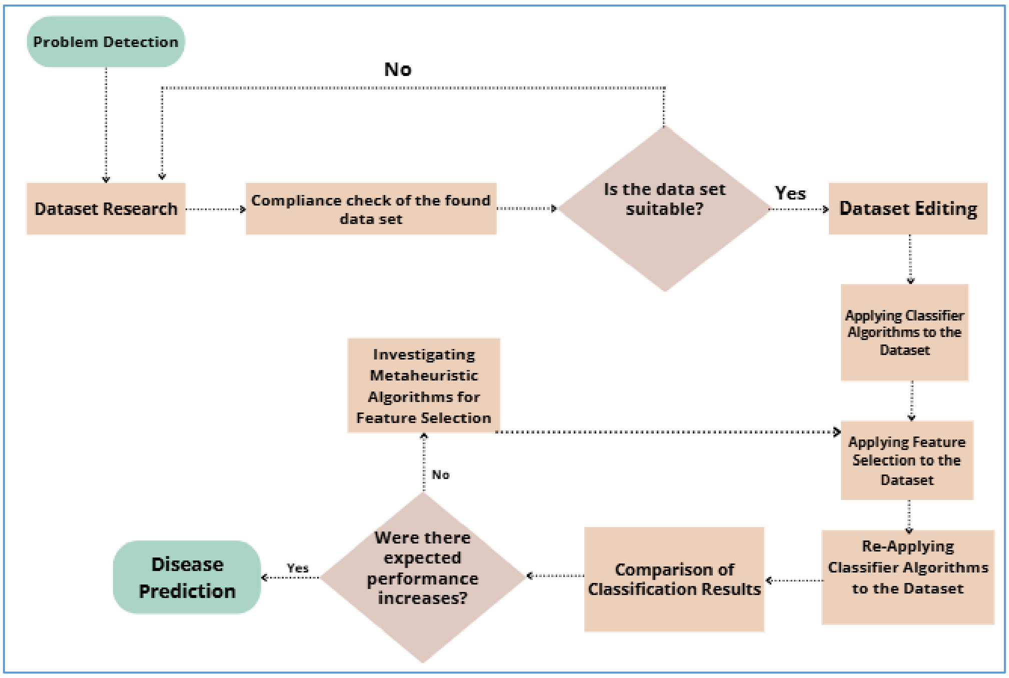
Problem Detection
No
Dataset Research
Compliance check of the found data set
Is the data set suitable?
Yes
Dataset Editing
Applying Classifier Algorithms to the Dataset
Investigating Metaheuristic Algorithms for Feature Selection
Applying Feature Selection to the Dataset
No
Re-Applying Classifier Algorithms to the Dataset
Comparison of Classification Results
Were there expected performance increases?
Yes
Disease Prediction

Figure 1. Workflow of the Study

4. Findings and Discussion

As the first step, selected classification algorithms were applied to the dataset, and their performance was evaluated with Weka Software. Subsequently, feature selection was performed on the dataset using CfsSubsetEval, to identify variables that significantly influence the disease. Using the CfsSubsetEval algorithm, 6 out of 32 independent variables were selected. These variables are "Family History Alzheimer's, Hypertension, Functional Assessment, Memory Complaints, Behavioral Problems, ADL." A new dataset was created with 6 independent variables and 1 dependent variable, and classification algorithms were run again. The performance results were compared to the previous (before feature selection) results. According to the performance results obtained, it was found that the CfsSubsetEval feature selection algorithm did not contribute to performance improvement. Results are provided in Table 2. The models performed consistently across training and test sets during the experiments, indicating no significant overfitting was observed.

Table 2. Comparative Results for Feature Selection

Classifier	Before Feature Selection	After Feature Selection	Classifier	Before Feature Selection	After Feature Selection
	Performance (%)	**Performance (%)**		**Performance (%)**	**Performance (%)**
BayesNet	93.39	84.03	MultiClass Classifier	83.71	81.10
NaiveBayes	84.83	80.40	MultiClass Classifier Updateable	83.38	80.87
Naive Bayes Multinominal Text	64.63	64.63	MultiScheme	64.63	64.63
NaiveBayes Updateable	84.83	80.40	Random Committee	92.41	82.41
LibSVM	64.63	85.38	Randomizable Filtered Classifier	56.95	77.85
Logistic	83.71	81.10	Random SubSpace	88.27	78.82
Multilayer Perceptron	82.41	86.41	Stacking	64.63	64.63
SGD	83.38	80.87	Vote	64.63	64.63
SGD Text	64.63	64.63	Weighted Instances Handler Wrapper	64.63	64.63
Simple Logistic	84.13	81.01	InputMapped Classifier	64.63	64.63
SMO	83.62	81.52	DecisionTable	95.57	87.85
Voted Perceptron	64.63	77.10	JRip	94.74	86.87
IBk	65.28	79.71	OneR	65.33	65.33
KStar	68.86	83.01	PART	90.27	86.73
LWL	69.61	80.40	ZeroR	64.63	64.63
Iterative Classifier Optimizer	94.74	86.87	DecisionStump	69.84	69.84
AdaBoostM1	93.90	85.01	HoeffdingTree	84.83	80.40
Attribute Selected Classifier	93.85	86.92	J48	94.88	86.92
Bagging	95.16	86.45	LMT	93.67	86.78
Classification Via Regression	93.25	87.24	RandomForest	94.69	84.69
CV Parameter Selection	64.63	64.63	RandomTree	80.64	79.85
Filtered Classifier	95.57	87.85	REPTree	94.69	87.01
LogitBoost	94.74	86.55			

As a next step of the study, two metaheuristic algorithms that are suitable for the existing dataset were selected and applied. These algorithms are the Genetic Algorithm and Particle Swarm Optimization algorithm. These algorithms were implemented and executed using the Python programming language. Using the Genetic Algorithm implemented in the Python programming language, 14 out of 32 independent variables were selected. These variables are "Ethnicity, Education Level, Physical Activity, Family History Alzheimer’s, Diabetes, Cholesterol Total, Cholesterol Triglycerides, MMSE, Functional Assessment, Memory Complaints, Behavioral Problems, ADL, Personality Changes, Difficulty

Completing Tasks". A new dataset was created with 14 independent variables and 1 dependent variable, and classification algorithms were run again. Results are presented in Table 3. The models performed consistently across training and test sets during the experiments, indicating no significant overfitting was observed.

Using the Particle Swarm Optimization implemented in the Python programming language, 14 out of 32 independent variables were selected. These variables are "Ethnicity, Education Level, BMI, Diet Quality, Family History Alzheimer's, Cardiovascular Disease, Diabetes, Cholesterol LDL, MMSE, Functional Assessment, Memory Complaints, Behavioral Problems, ADL, Forgetfulness". The classification algorithms were rerun using a fresh dataset that contained one dependent variable and fourteen independent variables. Results are presented in Table 3.

According to the obtained performance results, the highest performance rate of 95.57%, achieved before feature selection, remained unchanged after feature selection using the Genetic Algorithm and Particle Swarm Optimization, continuing to be the highest performance rate. This suggests that not all 32 variables in the Alzheimer's disease dataset are necessary, and only 14 of them may be sufficient to diagnose the disease. As a result, the time required for diagnosis is reduced, and the analysis speed is increased.

When comparing the Genetic Algorithm and Particle Swarm Optimization, both being meta-heuristic algorithms, it is necessary to make comparisons based on other performance rates since they provided the highest and identical performance rates. The Genetic Algorithm improved the performance of a total of 28 classifier algorithms, while the Particle Swarm Optimization algorithm improved the performance of a total of 33 classifier algorithms. Therefore, the algorithm that had the most significant impact on the dataset's performance is the Particle Swarm Optimization algorithm.

After applying Genetic Algorithms (GA) for feature selection, many classifiers show noticeable improvements in performance. For instance, the "RandomCommittee" classifier increases its accuracy from 92.41% (before feature selection) to 94.18% with GA, while "RandomForest" jumps from 94.69% to 95.39%. Similarly, "J48" also improves, reaching 95.11% after GA, compared to 94.88% before. These improvements suggest that Genetic Algorithms are particularly effective in enhancing the performance of more complex models, such as ensemble classifiers (e.g., Bagging, RandomForest), by optimizing the feature set.

On the other hand, Particle Swarm Optimization (PSO) generally shows similar or slightly better results compared to GA. In many cases, classifiers like "J48" and "Bagging" show no significant difference in performance with PSO, remaining close to their high pre-selection accuracy levels. However, classifiers like "RandomTree" and "KStar" perform slightly better with PSO than with GA, indicating that PSO may offer more suitable feature selection for certain models.

Table 3. Comparative Results of Genetic Algorithm and Particle Swarm Optimization Performances

Classifier	**Genetic Algorithm**	**Particle Swarm Optimization**	**Classifier**	**Genetic Algorithm**	**Particle Swarm Optimization**
	Performance (%)	**Performance (%)**		**Performance (%)**	**Performance (%)**
BayesNet	93.85	93.90	MultiClass Classifier	84.03	83.94
NaiveBayes	85.34	84.96	MultiClass Classifier Updateable	83.43	84.13
Naive Bayes Multinominal Text	64.63	64.63	MultiScheme	64.63	64.63
NaiveBayes Updateable	85.34	84.96	Random Committee	94.18	93.81
LibSVM	64.77	71.19	Randomizable Filtered Classifier	65.05	67.89

Table 3. Continued.

Classifier	**Genetic Algorithm**	**Particle Swarm Optimization**	**Classifier**	**Genetic Algorithm**	**Particle Swarm Optimization**
	Performance (%)	Performance (%)		Performance (%)	Performance (%)
Logistic	84.03	83.94	Random SubSpace	88.69	87.48
Multilayer Perceptron	85.57	84.69	Stacking	64.63	64.63
SGD	83.43	84.13	Vote	64.63	64.63
SGD Text	64.63	64.63	Weighted Instances Handler Wrapper	64.63	64.63
Simple Logistic	83.99	84.36	InputMapped Classifier	64.63	64.63
SMO	83.89	83.66	DecisionTable	95.57	95.57
Voted Perceptron	64.63	70.03	JRip	94.88	94.83
IBk	73.89	71.47	OneR	65.33	65.33
KStar	80.78	77.66	PART	92.55	91.67
LWL	70.03	69.93	ZeroR	64.63	64.63
Iterative Classifier Optimizer	94.74	94.74	DecisionStump	69.84	69.84
AdaBoostM1	93.90	93.90	HoeffdingTree	85.34	84.96
Attribute Selected Classifier	93.85	93.85	J48	95.11	95.16
Bagging	95.16	95.20	LMT	94.18	93.99
Classification Via Regression	94.64	94.32	RandomForest	95.39	95.20
CV Parameter Selection	64.63	64.63	RandomTree	85.24	86.83
Filtered Classifier	95.57	95.57	REPTree	94.88	94.83
LogitBoost	94.74	94.74			

Table 4 summarizes the impact of the feature selection methods on our highest performing classifiers.

Table 4. Highest performance classifiers and the impact of feature selection methods on their performance.

Classifier	**Performance Before Feature Selection (%)**	**CfsSubsetEval (%)**	**Genetic Algorithm (%)**	**Particle Swarm Optimization (%)**
FilteredClassifier	95.57	87.85	95.57	95.57
Bagging	95.16	86.45	95.20	95.20
RandomForest	94.69	84.69	95.39	95.20
J48	94.88	86.92	95.11	95.16
DecisionTable	95.57	87.85	95.57	95.57
JRip	94.74	86.87	94.88	94.83
RandomTree	80.64	79.85	85.24	86.83
KStar	68.86	83.01	80.78	77.66
MultilayerPerceptron	82.41	86.41	85.57	84.69

As can be seen from Table 4, the classifiers *FilteredClassifier*, *DecisionTable*, and *Bagging* achieved performances of over 95% after feature selection, using both GA and PSO. These classifiers maintained their performance even after reducing the number of features from 32 to 14. Though *CfsSubsetEval* resulted in a performance decrease for most of the classifiers in Table 4, coupling it with GA and PSO maintained or slightly improved performance. For example, the table clearly reveals that *RandomForest* improved from 94.69% to 95.39% with GA, and to 95.20% with PSO. *RandomTree* also showed a significant improvement with PSO, increasing its performance from 80.64% to 86.83%.

Rabie El Kharoua's Alzheimer's Disease dataset has been employed in several articles. Suñé (2025) [52] used classical machine learning techniques such as K-Nearest Neighbors (KNN), Logistic Regression, Support Vector Machines, Neural Networks, Decision Trees, and Random Forest, reporting the highest accuracy of 93.49% with the Decision Tree classifier. In another recent study, Patel et al. (2025) [53] experimented with several classifiers, including Logistic Regression, SVM, Decision Tree, Random Forest, and ANN. After testing across six different feature selection scenarios, they achieved the best result of 95% accuracy with Random Forest using selected numerical and categorical features. Sendhil et al. (2025) [54] compared SVM, GBM, Logistic Regression, KNN, and Random Forest using the same dataset, with Random Forest again performing best at 92% accuracy. Jevin & Umamageswari (2024) [55] applied a deep learning approach using a ABO-2layer CNN and achieved 94.56% accuracy, emphasizing the importance of hybrid deep learning models in medical prediction tasks. Airlangga (2024) [56] evaluated three deep learning architectures—MLP, CNN, and LSTM—on the dataset derived from the same source and found CNN to outperform others with an 88.65% average accuracy. Compared to these studies, our approach achieved a maximum accuracy of 95.57% using only 14 features selected via metaheuristic algorithms (Genetic Algorithm and Particle Swarm Optimization). This not only surpasses most prior studies in predictive accuracy but also demonstrates greater efficiency by reducing model complexity and dimensionality.

The findings of this study could be translated into clinical practice by streamlining diagnostic workflows in hospitals using a reduced feature set. By identifying the most relevant features, the diagnostic process could become faster, more cost-effective, and less resource-intensive, as fewer tests or measurements would be required.

Overall, both GA and PSO contribute to improved classification performance, but PSO seems to give a slight edge in some cases, particularly for classifiers that initially have lower performance. Overall, the best results are achieved with Bagging and FilteredClassifier, both reaching near 95% accuracy, with Particle Swarm Optimization providing the slight edge in enhancing their performance even further. In this case, it has been demonstrated that better results are achieved by selecting only the relevant features, as the classifiers perform more efficiently with the optimized feature sets. Since the best results were obtained with PSO, the selected and unselected features are visualized and presented below. Figure 2 illustrates a visual of the features selected in the PSO algorithm that gave the best result after applying Filtered Classifier with 14 features. Selected features have been illustrated with green and unselected features have been illustrated with blue color.

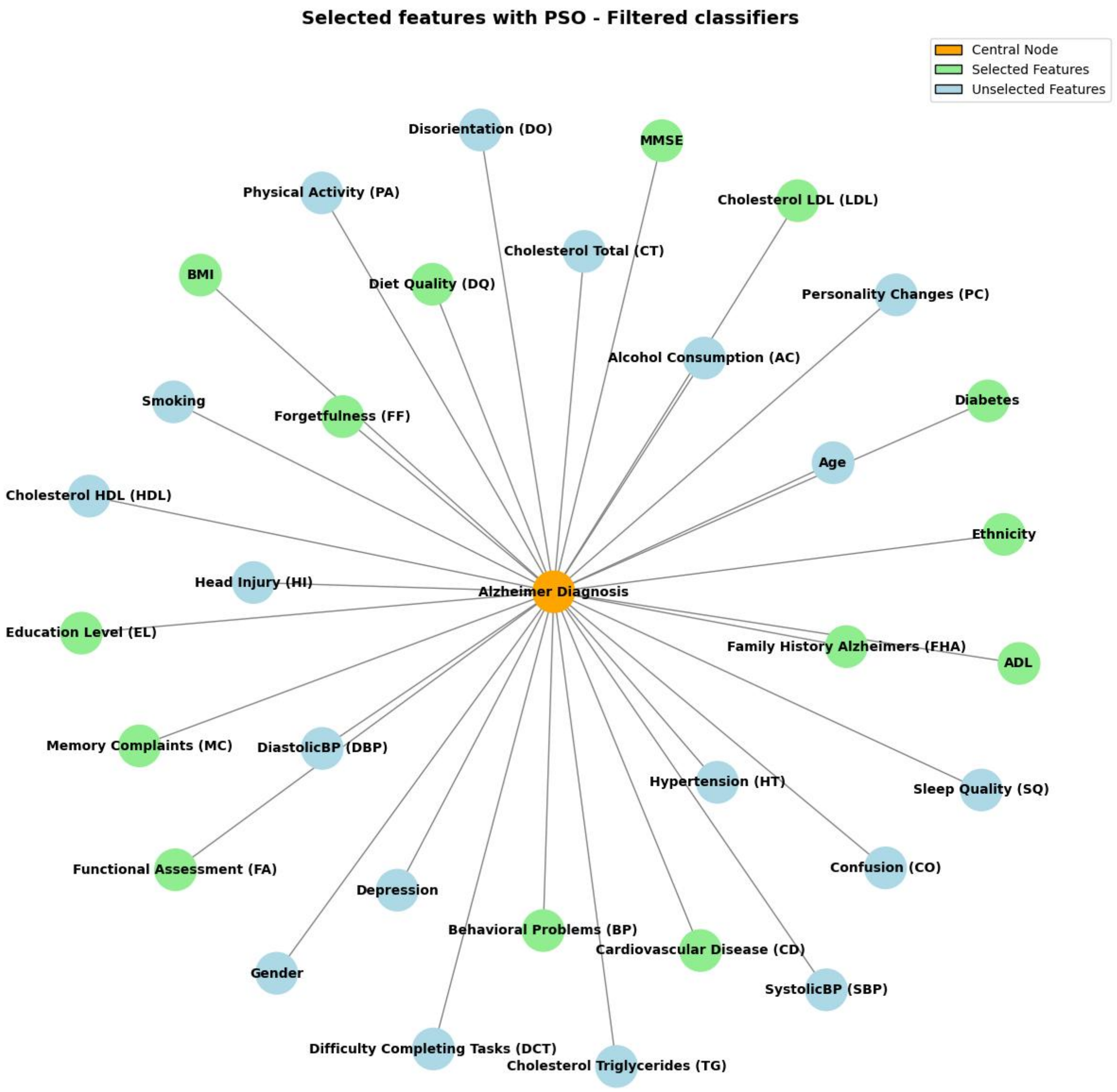

Figure 2. Selected features with best performance in diagnosis.

5. Conclusion

The aim of the study was to reduce the identification and data analysis time by selecting the minimum number of variables required for the diagnosis of Alzheimer’s disease, which, if not detected early, can negatively impact individuals' life trajectories and ultimately lead to death in the final stage. This approach also aims to enable the disease to be identified at the earliest possible stage.

In this study, a dataset containing 32 independent and 1 dependent variable, previously measured, was utilized. Then to reduce the number of variables in the dataset, to perform a performance comparison, the dataset was first run with 45 classifier algorithms, and performance results were obtained. Subsequently, CfsSubsetEval, was selected for feature selection and applied to the dataset. As a result of this application, a new dataset was created with 6 independent and 1 dependent variable, and this dataset was again run with 45 classifier algorithms.

Due to a significant drop in performance, it was decided to perform feature selection using meta-heuristic algorithms, considering previous studies. The Genetic Algorithms and Particle Swarm

Optimization algorithms were used to perform feature selection on the dataset in Python programming language. Although they are different from each other, both algorithms selected 14 independent and 1 dependent variable. Based on the selection of both algorithms, datasets were created and each was run again with 45 classifier algorithms. Finally, the results were compared. The highest performance rate before feature selection was 95.57%. After feature selection, both the Genetic Algorithms and Particle Swarm Optimization produced the same highest rate of 95.57%. This indicates that to achieve the highest performance, 14 variables, not 32, are sufficient. Since both algorithms produced the same rate, it is important to consider the increases in performance in other classifier algorithms. The Genetic Algorithms resulted in a performance increase in 28 classifiers, while Particle Swarm Optimization resulted in a performance increase in 33 classifiers. Therefore, it can be said that the best result for the dataset was provided by Particle Swarm Optimization. This study demonstrates that the disease can be detected more quickly and accurately with fewer variables, as the highest performance is achieved with 14 variables instead of 32.

Based on the results of our study, there could be several recommendations for future work and limitations to consider. Future research can focus on testing the proposed approach on larger and more diverse datasets to ensure generalizability and robustness across different populations. For this reason, future studies are planned to compare the results obtained with the ready-made dataset by collecting a real-world dataset. Additionally, exploring other metaheuristic algorithms, such as Ant Colony Optimization or Simulated Annealing, could provide a further insight into the effectiveness of alternative feature selection methods. Despite its promising findings, this study is limited by its reliance on a single dataset, which may not fully capture the variability present in real-world scenarios. Addressing this limitation in future research will help strengthen the applicability and reliability of the proposed methods in clinical practice. Finally, investigating the impact of feature selection on real-time clinical applications, such as integrating these methods into diagnostic tools or wearable devices, would also provide an invaluable future research direction.

Ethical statement

The data is sourced from an open-access database, so there is no need for an ethics committee's evaluation.

Acknowledgment

We are grateful to the anonymous referees for substantial suggestions that improved the quality of our study.

Conflict of interest

The authors declare no competing interests.

Authors' Contributions

S. Turkusay: Software, Formal analysis, Data curation, Writing-original draft. M. Oturakci, E. Ekinci, D. T. Eliiyi: Supervision, Conceptualization, Methodology, Writing - review & editing.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

References

[1] Kumar, A., Sidhu, J., Goyal, A., Tsao, J. W., "Alzheimer disease", *StatPearls*, StatPearls Publishing, 2023.

[2] T.C. Sağlık Bakanlığı, "Dünya Alzheimer Günü", 21 Eylül 2024 [Online]. Available: https://hsgm.saglik.gov.tr/tr/haberler-16/21-eylul-2024-dunya-alzheimer-gunu.html

[3] Porsteinsson, A. P., Isaacson, R. S., Knox, S., Sabbagh, M. N., Rubino, I., "Diagnosis of early Alzheimer's disease: clinical practice in 2021", *The journal of prevention of Alzheimer's disease*, 8, 371-386, 2021.

[4] Ding, Y., Zhou, K., Bi, W., "Feature selection based on hybridization of genetic algorithm and competitive swarm optimizer", *Soft Computing*, 24, 11663–11672, 2020.

[5] Anirudha, R. C., Kannan, R., Patil, N., "Genetic algorithm-based wrapper feature selection on hybrid prediction model for analysis of high-dimensional data", *Proceeding of 9th International Conference on Industrial and Information Systems (ICIIS)*, IEEE, pp. 1–6, 2014.

[6] Mirzaei, S., et al., "Two-stage feature selection of voice parameters for early Alzheimer's disease prediction", *IRBM*, 39(6), 430–435, 2018.

[7] Neelaveni, J., Devasana, M. S. G., "Alzheimer disease prediction using machine learning algorithms", *Proceeding of 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, IEEE, pp. 101–104, 2020.

[8] Saputra, R. A., et al., "Detecting Alzheimer's disease by the decision tree methods based on particle swarm optimization", *Journal of Physics: Conference Series*, IOP Publishing, 012025, 2020.

[9] Ramaswamy, R., Kandhasamy, P., Palaniswamy, S., "Feature selection for Alzheimer's gene expression data using modified binary particle swarm optimization", *IETE Journal of Research*, 69(1), 9–20, 2023.

[10] Divya, R., Shantha Selva Kumari, R., Alzheimer's Disease Neuroimaging Initiative, "Genetic algorithm with logistic regression feature selection for Alzheimer's disease classification", *Neural Computing and Applications*, 33(14), 8435–8444, 2021.

[11] Buyrukoğlu, S., "Early detection of Alzheimer's disease using data mining: Comparison of ensemble feature selection approaches", *Konya Journal of Engineering Sciences*, 9(1), 50–61, 2021.

[12] Noroozi, Z., Orooji, A., Erfannia, L., "Analyzing the impact of feature selection methods on machine learning algorithms for heart disease prediction", *Scientific Reports*, 13(1), 22588, 2023.

[13] Hassouneh, A., et al., "Feature importance analysis and machine learning for Alzheimer's disease early detection: Feature fusion of the hippocampus, entorhinal cortex, and standardized uptake value ratio", *Digital Biomarkers*, 8(1), 59–74, 2024.

[14] Kaggle, "Alzheimer's Disease Dataset" [Online]. Available: https://www.kaggle.com/datasets/rabieelkharoua/alzheimers-disease-dataset?select=alzheimers_disease_data.csv

[15] Soofi, Aized Amin, Awan, Arshad, "Classification techniques in machine learning: applications and issues", *Journal of Basic & Applied Sciences*, 13, 459–465, 2017.

[16] Sahoo, G., Kumar, Yugal, “Analysis of parametric & non parametric classifiers for classification technique using WEKA”, *International Journal of Information Technology and Computer Science (IJITCS)*, 4(7), 43, 2012.

[17] Surya, P. P., Seetha, L. V., Subbulakshmi, B., “Analysis of user emotions and opinion using multinomial naive Bayes classifier”, *Proceeding of 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, pp. 410–415, 2019.

[18] Abdiansah, A., Wardoyo, R., “Time complexity analysis of support vector machines (SVM) in LibSVM”, *International Journal of Computer Applications*, 128(3), 28–34, 2015.

[19] Bharati, S., Rahman, M. A., Podder, P., “Breast cancer prediction applying different classification algorithms with comparative analysis using WEKA”, *Proceeding of 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, IEEE, pp. 581–584, 2018.

[20] Alaoui, S. S., Farhaoui, Y., Aksasse, B., “Classification algorithms in data mining”, *International Journal of Tomography & Simulations*, 31(4), 34–44, 2018.

[21] Stupka, R. J., et al., “Exploring attribute selection and classification methods for predicting heart disease”, *Proceeding of International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, pp. 701–706, 2022.

[22] Mahmood, D. Y., Hussein, M. A., “Intrusion detection system based on K-star classifier and feature set reduction”, *IOSR Journal of Computer Engineering (IOSR-JCE)*, 15(5), 107–112, 2013.

[23] Dou, Y., Meng, W., “Comparative analysis of WEKA-based classification algorithms on medical diagnosis datasets”, *Technology and Health Care*, 31(S1), 397–408, 2023.

[24] Khosravi, K., et al., “Predictive modeling of selected trace elements in groundwater using hybrid algorithms of iterative classifier optimizer”, *Journal of Contaminant Hydrology*, 242, 103849, 2021.

[25] Thilagaraj, T., Sengottaiyan, N., “Implementation of classification algorithms in educational data using WEKA tool”, *International Journal of Computer Sciences and Engineering*, 7(5), 1254–1257, 2019.

[26] WEKA SourceForge, “AttributeSelectedClassifier” [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/classifiers/meta/AttributeSelectedClassifier.html

[27] WEKA SourceForge, “ClassificationViaRegression” [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/classifiers/meta/ClassificationViaRegression.html

[28] WEKA SourceForge, “CVParameterSelection” [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/classifiers/meta/CVParameterSelection.html

[29] WEKA SourceForge, “FilteredClassifier” [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/classifiers/meta/FilteredClassifier.html

[30] WEKA SourceForge, “LogitBoost” [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/classifiers/meta/LogitBoost.html

[31] WEKA SourceForge, “MultiClassClassifier” [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/classifiers/meta/MultiClassClassifier.html

[32] WEKA SourceForge, "MultiClassClassifierUpdateable" [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/classifiers/meta/MultiClassClassifierUpdateable.html

[33] WEKA SourceForge, "MultiScheme" [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/classifiers/meta/MultiScheme.html

[34] Niranjan, A., et al., "ERCR TV: Ensemble of random committee and random tree for efficient anomaly classification using voting", *Proceeding of 3rd International Conference for Convergence in Technology (I2CT)*, IEEE, pp. 1–5, 2018.

[35] Asaju, L. B., et al., "Intrusion detection system on a computer network using an ensemble of randomizable filtered classifier, K-nearest neighbor algorithm", *FUW Trends in Science & Technology Journal*, 2(1), 550–553, 2017.

[36] Lasota, T., Łuczak, T., Trawiński, B., "Investigation of random subspace and random forest methods applied to property valuation data", in: *Computational Collective Intelligence. Technologies and Applications* (Ed. N.T. Nguyen et al.), Berlin, Springer Berlin Heidelberg, 2011, ch.3, pp. 142–151.

[37] Baskin, I. I., Marcou, G., Horvath, D., Varnek, A., "Cross-Validation and the Variable Selection Bias", in: *Tutorials in Chemoinformatics* (Ed. A. Varnek), Chichester, John Wiley & Sons, 2017, ch.10, pp. 163–173.

[38] WEKA SourceForge, "Vote" [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/classifiers/meta/Vote.html

[39] Kargar, K., Safari, M. J. S., Khosravi, K., "Weighted instances handler wrapper and rotation forest-based hybrid algorithms for sediment transport modeling", *Journal of Hydrology*, 598, 126452, 2021.

[40] WEKA SourceForge, "InputMappedClassifier" [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/classifiers/misc/InputMappedClassifier.html

[41] Jankovic, R., "Classifying cultural heritage images by using decision tree classifiers in WEKA", Proceeding of 1st International Workshop on Visual Pattern Extraction and Recognition for Cultural Heritage Understanding Co-located with 15th Italian Research Conference on Digital Libraries (IRCDL 2019), Pisa, Italy, pp. 119–127, 2019.

[42] WEKA SourceForge, "JRip" [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/classifiers/rules/JRip.html

[43] Nasa, C., Suman, S., "Evaluation of different classification techniques for web data", *International Journal of Computer Applications*, 52(9), 34–40, 2012.

[44] Aher, S. B., Lobo, L. M. R. J., "Data mining in educational system using WEKA", *Proceeding of International Conference on Emerging Technology Trends (ICETT)*, Foundation of Computer Science, pp. 20–25, 2011.

[45] Kawelah, W., Abdala, A., "A comparative study on machine learning tools using WEKA and RapidMiner with classifier algorithms C4.5 and decision stump for network intrusion detection", *European Academic Research*, 7(2), 852–861, 2019.

[46] Lavanya, K., et al., "Handwritten digit recognition using Hoeffding tree, decision tree and random forests—a comparative approach", *Proceeding of International Conference on Computational Intelligence in Data Science (ICCIDS)*, IEEE, pp. 1–6, 2017.

[47] Rajalakshmi, A., Vinodhini, R., Bibi, K. F., “Data discretization technique using WEKA tool”, *International Journal of Science, Engineering and Computer Technology*, 6(8), 293, 2016.

[48] Kumar, V., Minz, S., “Feature selection”, *SmartCR*, 4(3), 211–229, 2014.

[49] Velmurugan, T., Anuradha, C., “Performance evaluation of feature selection algorithms in educational data mining”, *Performance Evaluation*, 5(2), 92–106, 2016.

[50] Rostami, M., Berahmand, K., Forouzanadeh, S., “A novel community detection-based genetic algorithm for feature selection”, *Journal of Big Data*, 8(1), 2, 2021.

[51] Xue, B., Zhang, M., Browne, W. N., “Particle swarm optimization for feature selection in classification: A multi-objective approach”, *IEEE Transactions on Cybernetics*, 43(6), 1656–1671, 2012.

[52] Suñé, F. G., “Machine learning techniques for Alzheimer prediction”, Master’s thesis, Universitat de Barcelona, 2025.

[53] Patel, P., Patel, K., Thakkar, K., Goel, P., Valani, P., Vadhavana, V., “Alzheimer’s disease detection using various machine learning algorithms”, *Proceeding of 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI-2025)*, IEEE, pp. 1048–1054, 2025.

[54] Sendhil, R., Patibandha, N., Shah, P., Mandal, A., “Comparative Evaluation of Machine Learning Techniques for Early Alzheimer’s Detection: Challenges and Insights”, *Proceeding of International Conference on Multi-Agent Systems for Collaborative Intelligence (ICMSCI)*, IEEE, pp. 1050–1058, 2025.

[55] Jevin, J. A., Umamageswari, A., “Alzheimer’s Disease Prediction using an Artificial Butterfly Optimizer with a Two-Layer CNN Approach”, *Proceeding of 4th International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, IEEE, pp. 1–6, December 2024.

[56] Airlangga, G., “Advancing Alzheimer’s Diagnosis: A Comparative Analysis of Deep Learning Architectures on Multidimensional Health Data”, *Jurnal Informatika Ekonomi Bisnis*, pp. 810–814, 2024.