The effect of feature selection with optimization on taxi fare prediction

Amany A. Naim[1], Asmaa Hekal Omar[1], Asmaa A. Ibrahim[1], Asmaa Mohamed[1], Naglaa M. Mostafa[1,2]
[1]Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, Egypt
[2]Department of Computer and Information Science, Applied College, Taibah University, Kingdom of Saudi Arabia

Article Info

Article history:

Received Oct 4, 2024
Revised Dec 20, 2024
Accepted Mar 9, 2025

Keywords:

Fare prediction
Feature selection
Genetic algorithm
Machine learning
Optimization
Predictive analysis
Supervised learning

ABSTRACT

Feature selection plays a key influence in machine learning (ML); the main objective of feature selection is to eliminate irrelevant and redundant variables in different classification problems to improve the performance of the learning algorithms. Classification accuracy is improved by reducing the number of selected features. Many real-world problems, such as taxi fare can be predicted by ML. This paper proposes feature selection using genetic algorithm (GA) optimization to predict taxi fare. Experiments are performed on real datasets of taxi fare, and this paper uses eight classifiers to evaluate the selected features. The performance of the classifiers is assessed using various performance metrics. The results are compared with feature selection without optimization. The proposed method records high classification accuracy when evaluated by three types of classifiers (random forest, AdaBoost, and Gradient Boost). The results indicate that the prediction accuracy of the proposed method is 99.7% on taxi fare dataset.

Corresponding Author:

Asmaa Mohamed
Department of Mathematics, Faculty of Science, Al-Azhar University
Cairo, Egypt
Email: asmaamohamed89@azhar.edu.eg

1. INTRODUCTION

Taxis play an important role in urban public transportation. So, accurate taxi fare forecasting is crucial for service providers and customers alike, as it facilitates improved planning, pricing, and budgeting for transportation requirements. The increasing use of data-driven methods and developments in machine learning (ML) and optimization techniques have enabled the development of more accurate and economical models for predicting taxi fares [1]. Feature selection and predictive modelling are essential components of ML systems. As modern datasets increase in size and complexity, it becomes more challenging to accurately identify meaningful inputs and map them to desired outputs [2].

This paper examines the methods used in feature engineering and classification modelling for supervised learning challenges. The feature selection domain investigates various categories, such as filter methods, wrapper methods, and embedding techniques [3]. Filter approaches [4] utilise inherent data characteristics, such as inter-feature correlation, to evaluate the significance of incoming data without considering the model. Wrapper approaches [5] assess feature subsets by measuring the performance of a selected classifier. Embedded approaches incorporate variable selection directly into the process of constructing the model. In addition to these methods, metaheuristic search algorithms such as genetic algorithm (GA) can effectively explore the combinatorial feature space using biologically inspired optimization techniques [6].

The paper examines a wide range of inductive biases in categorisation learning. Instance-based methodologies, such as k-nearest neighbor (KNN) models, store training examples and evaluate fresh samples based on their similarity to previously stored cases. Decision trees (DT) divide the feature space into smaller sections by finding splits aligned with the axes and maximising purity. Support vector machines (SVM) utilise kernelised feature spaces to discover the most effective separation hyperplanes. Ensemble methods, such as random forests, utilise many DT to enhance forecast accuracy by reducing variation.

This research presents a system that utilises GA to determine the most suitable attributes for precise fare classification. Utilising GA-guided search has demonstrated encouraging outcomes for reducing dimensionality in ML datasets [7], [8]. The main concept is to utilise the GA to systematically investigate various combinations of input factors inside taxi data and identify the ones that yield the most accurate fare prediction model. Utilising this subset of predictive features rather than the original variables offers several benefits, including increased model interpretability, quicker training periods, and enhanced accuracy [9]. This paper presents two distinct contributions. GA uses binary encoding, and specialised operators is constructed to explore the feature space. The classification accuracy on a held-out set determines the fitness score for evolving informative feature subsets. Furthermore, the GA-selected reduced features are used to construct and analyse various machine-learning models for fare classification, assessing the enhancement in generalisation compared to traditional full-feature methods.

The results illustrate the efficacy of employing GA-based feature selection to improve the performance of ML models for fare prediction. The GA identifies optimal feature subsets, resulting in simpler, faster, and more accurate models than those obtained using brute-force methods. These findings have consequences for developing ML systems that are ready for production and can accurately estimate taxi/ride-sharing fares. The subsequent of the paper is organised as follows: it consists of four sections: section 2 discusses the methodologies which are used in this study; section 3 explicates the proposed method; and section 4 shows the experimental results of the experiment and discussion. Section 5 summarises the results of this work and draws the conclusion.

2. METHOD

ML provides various effective methods for classification tasks, each with unique advantages and uses. This overview examines various widely used techniques, such as KNN, SVM with linear and radial basis function (RBF) kernels, DT, random forests, logistic regression, GradientBoost, and AdaBoost. These algorithms encompass a range of methodologies, ranging from essential instance-based learning to complex ensemble methods. Comprehending these methods' fundamental principles is essential to select the most appropriate algorithm for a certain task and interpret its outcomes effectively.

2.1. Feature selection

Currently, practitioners and researchers handle extensive data sets comprising hundreds to several thousand attributes. Feature selection is a dimensionality reduction method aimed at identifying a subset of relevant characteristics from those of lesser significance, while preserving optimal predictive performance. This task is justified by multiple factors: reduced computing expense for training and forecasts, enhanced predictive strength, and increased interpretability. However, distinguishing between relevant and non-relevant features is not trivial, so many selection methods exist [6].

The selection of method significantly impacts the behavior of the final model; therefore, ML developers must comprehend the employed methodologies to effectively communicate their judgments to stakeholders, which include regulators. Feature selection for supervised learning tasks can be categorized into: i) filtering, ii) wrapper, or iii) built-in approaches [7]. In the literature, several metaheuristic algorithms have been developed and used to solve feature selection problems: GA [8], simulated annealing (SA) [9], ant colony optimization (ACO) [10], differential evolution (DE) [11], particle swarm optimization (PSO) [12], artificial bee colony (ABC) [13], and firefly algorithm (FFA) [14].

2.2. Genetic algorithm

The GA, derived from the evolutionary process of genetics, was proposed by Holland in 1975 [15]. It is a globally adaptable heuristic search approach commonly employed to identify an optimal solution to a specified problem [16], [17]. Every chromosome in the GA signifies a solution to the specified problem.

All chromosomes comprise the population, and the number of chromosomes within the population is termed the population size. The fundamental components of the GA are the crossover and mutation operators, which produce offspring by combining genetic information from several parents and maintain genetic diversity throughout generations through the probabilistic modification of particular genes [18]. The researchers have proposed different variants of GA that have been extensively used in diverse fields. GA can be categorized into six primary types depending on various coding schemes: binary, octal, hexadecimal, permutation, value-based, and tree-based [8], [19].

The GA is a method for addressing optimization issues grounded in the biological principles of evolution, specifically natural selection. A GA is an effective variant of the standard evolutionary algorithm that selects random solutions from the current population at each step, designates them as parents, and utilizes them to generate the next generation of offspring through a series of processes. The researchers have introduced several iterations of GA that have been widely employed in various fields. GA can be categorized into six principal types based on distinct coding schemes: binary, octal, hexadecimal, permutation, value-based, and tree-based. They employ biological operations, specifically reproduction, selection, crossover, and mutation [20]. The fundamental progression of a GA solution is illustrated in Algorithm 1.

Algorithm 1. Generic GA

```
Start
Initialise population randomly (say P)
Define fitness function of the problem
Determine the fitness of the population
While !Converging or Optimum not achieved do
    Parent selection from  population
    Crossover operation for new population generation
    Perform mutation on the new population
    Calculate fitness of new population
End
If optimum achieved, display the final result
Stop
```

The procedures of GA are as follows: i) a genetically appropriate representation of the solution domain in a computaionally suitable for computation, ideally binary representation (0 and 1) [21]; ii) a function to assess the efficacy of the solution or the population created (fitness function); and iii) data Initialization. Each potential answer is often depicted as a bit array. Arrays and other data structures can be utilized similarly [22].

The primary attribute of these genetic representations that renders them advantageous is the straightforward alignment of components, which facilitates crossover procedures. Utilization of variable-length representations is feasible; nevertheless, it complicates cross-implementation processes.

2.3. Classification in machine learning

Classification is a supervised ML technique in which the model endeavors to predict the accurate label for specified input data. Classification entails thorough training the model utilizing training data, subsequently assessing it with test data, and employing it for predictions on novel, unseen data. In ML classification, there exist two categories of learners: lazy learners and eager learners. Eager learners are ML algorithms that develop a model using the training dataset prior to predicting outcomes on subsequent datasets. Illustrations: logistic regression, SVM, DT, and artificial neural networks.

Conversely, lazy learners or instance-based learners do not promptly develop a model from the training data, which accounts for their lethargic characteristics. They retain the training data and, if a prediction is required, they locate the nearest neighbor by referencing the complete training dataset, resulting in a sluggish prediction process. Examples of this category include KNN and case-based reasoning [23]. There are various techniques for classification in ML, such as KNN, SVM, DT, random forest, logistics, AdaBoost, and GradientBoost.

2.3.1. K-nearest neighbor

This method's principle is to identify a certain number of training taxi fare samples nearest to the new point and forecast the label based on them. The quantity of samples may be a user-defined constant or fluctuate based on the local point density. The distance may be expressed in any metric unit of measurement. Various distance metrics are employed in KNN [24], including Manhattan, Euclidean, and edit distance, with Euclidean distance being the most favored. Nonetheless, it excels in a diverse array of categorization challenges.

2.3.2. Linear support vector machine

The linear SVM classifier is a ML technique frequently employed for binary classification applications. It works by finding the best hyperplane that separates the data points of different classes with the maximum margin. This means that it aims to create a decision boundary that maximises the distance between the closest points of each class. By doing so, it can effectively classify new data points based on their position relative to this decision boundary. The LinearSVM classifier is particularly useful when dealing with large datasets and high-dimensional feature spaces, as it can efficiently handle these scenarios while maintaining good generalisation performance. SVM have been widely utilized in diverse domains, including time series analysis and signal processing. According to the statistical learning hypothesis and the notion of structural risk minimization, SVM is less susceptible to overfitting and employs a linear function hypothesis space within a significantly higher dimensional feature space. Research indicates that SVMs outperform conventional artificial neural networks in addressing classification and regression challenges owing to their greater generalization capabilities [25].

2.3.3. RadiailSVM

Kernel SVM are resilient ML techniques utilized for classification and regression tasks. They have gained importance due to their ability to handle high-dimensional data and their effectiveness in solving complex problems. SVMs function by determining an optimal hyperplane that maximally differentiates between multiple classes of data points [26].

The RBF kernel is among the most often utilized kernels in SVM. A non-linear kernel enables SVM to proficiently classify data that is not linearly separable within the input space. The RBF kernel evaluates the similarity between two data points by assessing their distance from one another, utilizing a Gaussian distribution. This enables SVMs to identify intricate correlations and produce precise predictions, even when the decision boundary is non-linear. The RBF kernel has been effectively utilized across multiple fields, including image recognition, text classification, and bioinformatics.

2.3.4. Decision trees

A DT comprises two categories of components: i) leaf nodes that allocate class labels to observations and ii) internal nodes that delineate tests for certain qualities, accompanied by a branch and subtree for each outcome of the test. The tree classifies observations from top to bottom, extending from the base through its own method downward based on test results on internal nodes, until assigning a class label and reaching a leaf node. The tree is then built by algorithmic partitioning until these leaf nodes only contain instances of a single class or until no analysis provides an improvement [27].

2.3.5. Random forest

This approach employs supervised learning, facilitating both regression and classification tasks. Random forest outperforms a solitary DT as it comprises several DT that collectively enhance the prediction of the goal value. A compilation of trees yields a more precise result than an individual tree [28].

2.3.6. Logistic

Logistic regression is a method that applies the principles of linear regression to classification issues. The classification outcome is a value within the range [0, 1], understood as the probability $h(x)$ that the class of x is 1. The sigmoid function employed in logistic regression is the logistic function, as delineated in (1):

$$f(z) = \frac{1}{1+e^{-z}} \tag{1}$$

where z is of the form represented in (2):

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \tag{2}$$

where x_1 to x_n represent the values of the n attributes and β or to β_n represent the weights [29].

2.3.7. GradientBoost

The GradientBoost classifier is a powerful ML algorithm combining gradient boosting principles and DT. It is widely used for both classification and regression tasks, offering high accuracy and flexibility in handling complex datasets. By iteratively training weak learners and optimising the loss function, the GradientBoost classifier gradually improves its predictive performance, making it a popular choice in various domains such as finance, healthcare, and natural language processing [30].

2.3.8. Adaboost

Adaptive boosting is a ML technique that amalgamates weak and ineffective predictive algorithms to provide precise forecasts. The Adaboost method generates a classification by amalgamating many categories. Each rating possesses a weight, and a significant new rating is generated when these weights are aggregated. Despite being an inferior classification method, weak classification surpasses random prediction. A straightforward method to modify weak categories into functional classifications, each reliant on a singular quality, is an uncomplicated approach to their adaptation. This strategy does not necessitate an extensive database, given the prevalent use of databases [31].

2.4. Metrics for evaluating machine learning classification algorithms

Now that we have an idea about the different types of classification models, it is essential to choose the right evaluation metrics for these models. In this section, the most commonly used metrics: accuracy, TNR, precision, recall, F1 score, sensitivity, specificity, and G-mean are covered through (3) to (10), respectively. The confusion matrix is a visualisation tool frequently used in supervised learning. Each column of the matrix exemplifies a predicted class, whereas each row denotes events in the actual class [32]. Table 1 shows the confusion matrix, which encompasses real and predictable information about the classification system.

Table 1. Confusion matrix

Actual class	Predicted class	
	Predicted. class 1	Predicted. class 0
Actual. class 1	(True positive)	(False negative)
Actual. class 0	(False positive)	(True negative)

where:
True positive (TP)=the quantity of positive instances accurately identified by the system;
True negative (TN)=the quantity of negative instances accurately categorized by the system;
False negative (FN)=the quantity of negative instances incorrectly categorized as positive by the system;
False positive (FP)=the quantity of positive data incorrectly categorized as negative by the system.
Equation for the confusion matrix:

$$\text{Accuracy}=\frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{3}$$

$$\text{TNR}=\frac{\text{TN}}{\text{TN+FP}} \tag{4}$$

$$\text{precision}=\frac{\text{TP}}{\text{TP+FP}} \tag{5}$$

$$\text{Recall}=\frac{\text{TP}}{\text{TP+FN}} \tag{6}$$

$$\text{F1}-\text{score}=2\times\frac{\text{precision}\times\text{Recall}}{\text{precision+Recall}} \tag{7}$$

$$\left[\text{Sensitivity}=\frac{\text{TP}}{\text{TP + FN}}\right] \tag{8}$$

$$\left[\text{Specificity}=\frac{\text{TN}}{\text{TN+ FP}}\right] \tag{9}$$

$$\text{G-mean=sqrt(sensitivity+specifity)} \tag{10}$$

The correlation matrix is used to show the strong correlation between the variables. According to the matrix, all independent variables are important to the prediction variable because they all contribute to it. There are other analyses as well, but we used correlation analysis for feature selection.

3. METHOD

The primary aim of this research is to enhance the accuracy of predicting taxi fare amounts using ML techniques. An intelligent feature selection method using GA is proposed prior to model building to identify the optimal subset of input features and then train the classification models.

3.1. Dataset description

The dataset contains 8 features. The full list of features and descriptions is set out in Table 2. The data was collected from the Kaggle website. The dataset also had approximately 5 million rows of data. We have utilised almost 80,000 rows of data from the years 2009 to 2016 [33].

Table 2. Full list of features and description in the initial dataset

Feature name	Description and values
trip_duration	What was the duration of the journey?[in seconds]
distance_traveled	What was the distance traveled by the taxi?[in kilometers]
num_of_passengers	What is the number of passengers in the taxi?
fare	What is the base fare for the trip?[In Indian Rupees]
tip	What was the amount of tips received by the driver?[In Indian Rupees]
miscellaneous_fees	Were were any supplementary fees incurred during the journey?for example, tolls, convenience fees, goods, and services tax.[In Indian Rupees]
total_fare	The overall sum for the journey (this is your prediction objective) [in Indian Rupees]
surge_applied	Was surge pricing implemented?yes or no?

3.2. The proposed intelligent based model

The proposed method involves: an advanced feature selection approach to enhance the accuracy of taxi fare predictions. The method involves using a genetic algorithm (GA) to search for the optimal subset of features from the taxi dataset that are most predictive of the fare amount, ensuring that only the most relevant data points are utilized.

3.2.1. Feature selection method

Feature selection uses a GA, which is used to search for the optimal subset of features from the taxi dataset that are most predictive of the fare amount. The GA fitness function is designed to maximise classification accuracy. Only the features-the GA selects trip duration, distance traveled, number of passengers, and surge applied.

3.2.2. Genetic algorithm-based search

A GA based search is implemented to explore the feature space and select predictive features. Each chromosome in the GA represents a subset of features. Features are encoded as binary genes in the chromosome. An initial population of 50 random chromosomes representing different feature subsets is created. The fitness function guiding evolution is designed to maximise classification accuracy on a validation set. Over ten generations, crossover mutation operators are applied to retain and propagate fit chromosomes.

Algorithm Parameters: population size=50, generations=10, crossover rate=0.8, mutation rate=0.05, elitism parameter=5 (fittest chromosomes copied to next generation). Termination and feature subset selection: the algorithm is terminated after ten generations. The chromosome in the final generation with the highest classification accuracy determines the selected features. Based on the search, the features trip_duration, distance_traveled, num_of_passengers, and surge_applied are identified as the optimal set for fare prediction.

3.2.3. Classification model development

Classification model development is divided into four main parts The full proposed model is shown in Figure 1. It will be illustrated in full detail in the following sections:

a. Data splitting: the entire taxi dataset is split 80:20 into train and test sets for model building.
b. Model training: eight classification models are trained on the reduced training set with features identified by the GA with logistic regression, random forest, SVM, KNN, DT, AdaBoost, and Gradient Boosting.
c. Hyperparameter tuning through grid search is done for each model to optimise accuracy.
d. Evaluation: the trained models are tested on the 20% of test data held out. Calculated performance metrics include accuracy, precision, recall, and F1-score. Results are compared to baseline models trained on a complete feature set.

By selecting predictive signals upfront using GA, the method aims to build simpler, faster, and more accurate fare classifiers compared to no feature selection. The key results demonstrate a significant boost in accuracy, F1-score for random forest, AdaBoost, and Gradient Boosting trained on the GA-selected features. In particular, accuracy reaches 99.7%, precision reaches 99.59%, and recall reaches 100% for these models. The proposed approach of applying GA-based feature selection prior to model training leads to simpler and highly accurate models for predicting taxi fare compared to those with no feature selection. The GA successfully identifies the most relevant signals, improving generalisation and performance across multiple classifier architectures.

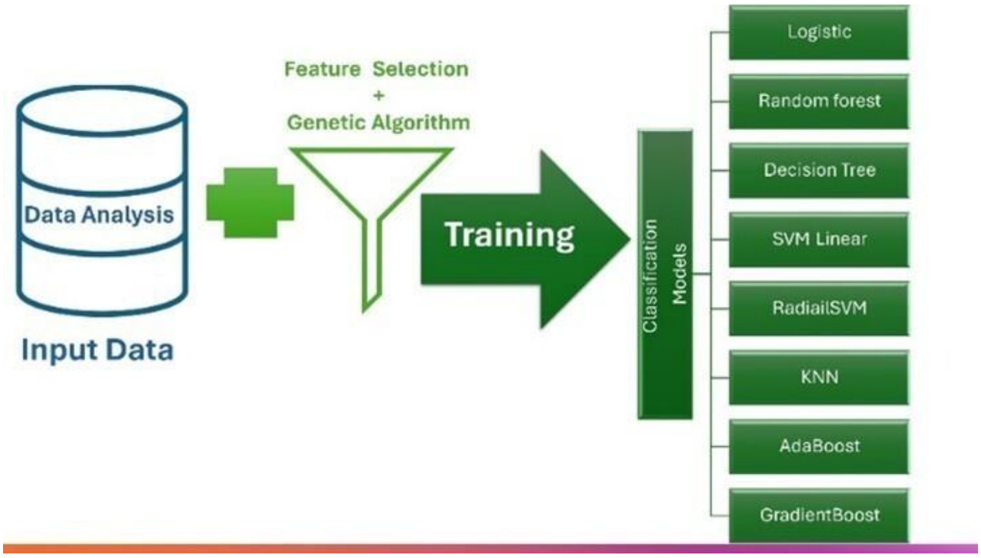


Figure 1. Proposed model for prediction of taxi fare data set

4. RESULTS AND DISCUSSION

This section validates the efficiency of using feature selection and ML based methods to predict taxi fare. All results are based on a set of standard evaluation metrics such as accuracy, precision, F-measure, recall, TNR, and G-mean. The distribution of dataset (trip_duration, distance_traveled, num_of_passengers, fare, tip, miscellaneous_fees, total_fare, surge_applied) is shown in Table 3 and Figure 2. The correlation matrix between dataset as shown in Figure 3. Table 3 shows the details and many characteristics for dataset such as count, mean, standard deviation, minimum, range of distribution (25%, 50%, and 75%), and maximum value for each dataset. For example, the count of trip_duration is 209673, the mean is 1173.18, the standard deviation is 4775.65, the minimum and maximum values are 0 and 86387 and 25 %, 50%, 75% from the data for trip_duration are 4460, 707, 1098 respectively.

Figure 3 shows the distribution of the dataset. The horizontal axis represents the features of the dataset, while the vertical axis represents the count of these features. For example, the minimum value for trip_duration is 0 and the maximum value is 86387. The highest distribution ratio is 707. The minimum value for distance_traveled is 0.02, and the maximum value is 57283.91. When the value is in the range [1.95, 5.73], the data dispersion is more pronounced. The start value for num_of_passengers is 0, and the end value is 9, but the highest distribution ratio is 1.

The experiments are performed to determine the classification performance measures (accuracy, precision, F-measure, recall, TNR, and G-mean) using eight classifiers (logistic, random forest, DT, LinearSVM, RadiailSVM, KNN, AdaBoost, and GradientBoost). This was done in two stages. The first stage received a classification without any improvement plan (without feature selection-optimization), and the second stage obtained a classification after feature selection- optimization. The results of experiments are shown in Tables 4 and 5 respectively.

Table 3. Analysis of dataset

	trip_ duration	distance_traveled	num_of_passengers	fare	tip	miscellaneous_ fees	total_fare	surge_applied
count	209673	2096730	209673	209673.	209673	209673	209673	209673
mean	1173.18	5.05	1.29	99.62	13.03	15.14	127.79	0.28
std	4775.65	125.22	0.93	85.60	20.37	12.55	98.80	0.45
min	0	0.02	0	0	0	-0.50	0	0
25%	4460	1.95	1	52.50	0	6.000	70.20	0
50%	707	3.20	1	75	9	9.75	101.70	0
75%	1098	5.73	1	116.25	20	26.45	152.25	1
max	86387	57283.91	9	4466.25	2500	435	4472.25	1

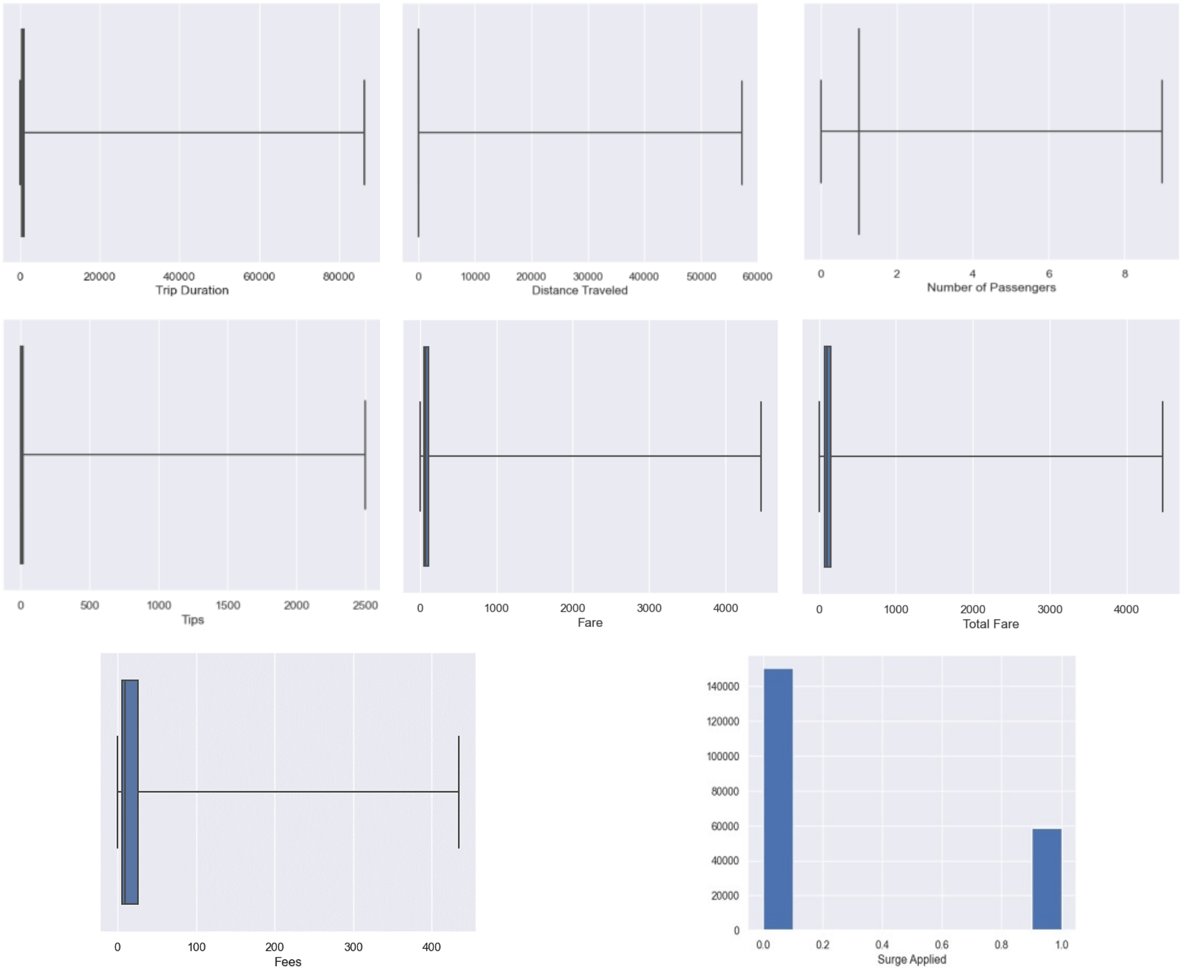
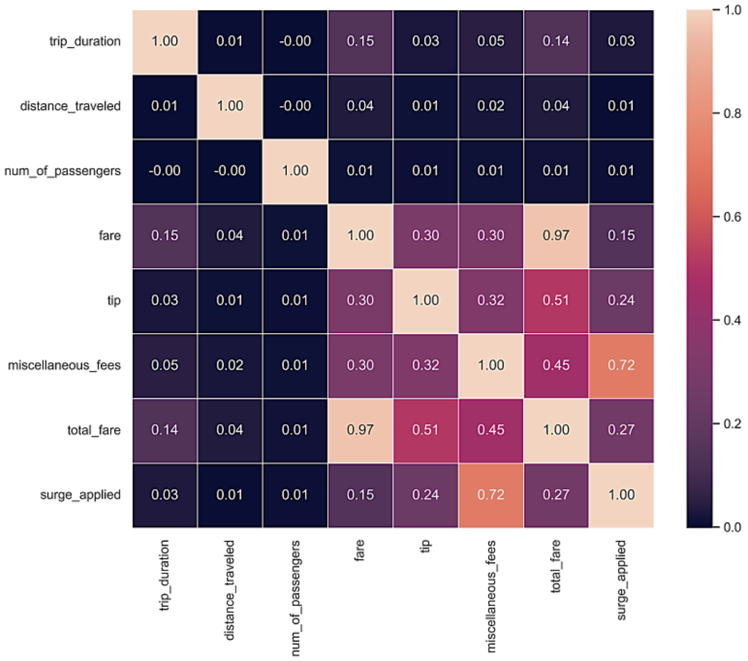


Figure 2. Distributions of dataset



	trip_duration	distance_traveled	num_of_passengers	fare	tip	miscellaneous_fees	total_fare	surge_applied
trip_duration	1.00	0.01	-0.00	0.15	0.03	0.05	0.14	0.03
distance_traveled	0.01	1.00	-0.00	0.04	0.01	0.02	0.04	0.01
num_of_passengers	-0.00	-0.00	1.00	0.01	0.01	0.01	0.01	0.01
fare	0.15	0.04	0.01	1.00	0.30	0.30	0.97	0.15
tip	0.03	0.01	0.01	0.30	1.00	0.32	0.51	0.24
miscellaneous_fees	0.05	0.02	0.01	0.30	0.32	1.00	0.45	0.72
total_fare	0.14	0.04	0.01	0.97	0.51	0.45	1.00	0.27
surge_applied	0.03	0.01	0.01	0.15	0.24	0.72	0.27	1.00

Figure 3. The correlation matrix between dataset

Table 4. The result of classification of original dataset without feature selection-optimization using eight classifiers and evaluating the performance

Classifier	Accuracy (%)	Precision (%)	F-measure (%)	Recall (%)	TNR (%)	G-mean (%)
Logistic	88.15	88.63	88.18	88.15	11.85	32.32
Random forest	95.46	95.49	95.47	95.46	4.54	20.82
DT	94.13	94.11	94.11	94.13	5.87	23.51
LinearSVM	83.25	83.3	82.52	83.25	16.75	37.34
RadiailSVM	70.97	71	70.01	70.97	29.03	45.39
KNN	76.18	75.94	74.16	76.18	23.82	42.60
AdaBoost	95.46	95.49	95.47	95.46	4.54	20.82
GradientBoost	94.13	94.11	94.11	94.13	5.87	23.51

Table 5. The result of classification of dataset with feature selection-optimization using eight classifier and evaluating the performance

Classifier	Accuracy (%)	Precision (%)	F-measure (%)	Recall (%)	TNR (%)	G-mean (%)
Logistic	94.96	95.92	96.51	97.12	2.90	16.77
Random forest	99.70	99.59	99.79	100	0	0
DT	99.41	99.18	99.59	100	0	0
LinearSVM	94.96	95.92	96.51	97.12	2.90	16.77
RadiailSVM	71.81	71.81	83.59	100	0	0
KNN	87.54	90	91.46	92.98	7.02	25.56
AdaBoost	99.70	99.59	99.79	100	0	0
GradientBoost	99.70	99.59	99.79	100	0	0

The results in Table 4 and Figure 4 show a comparison for the confusion matrix by eight classifiers without feature selection optimization. They show that accuracy 95.46%, precision 95.49%, F-measure 95.47%, recall 95.46% which are higher performance using random forest and AdaBoost classifiers than the others. In Figure 4, the horizontal axis represents the eight classifiers that applied to the original dataset (without feature selection optimization), while the vertical axis represents the performance of these classifiers. The results in Table 5 and Figure 5 show that accuracy 99.70%, precision 99.59%, F-measure 99.79%, recall 100% which are higher performance using random forest, AdaBoost and GradientBoost classifiers than the others. In Figure 5, the horizontal axis represents the eight classifiers that applied to the original dataset (with feature selection optimization), while the vertical axis represents the performance of these classifiers.

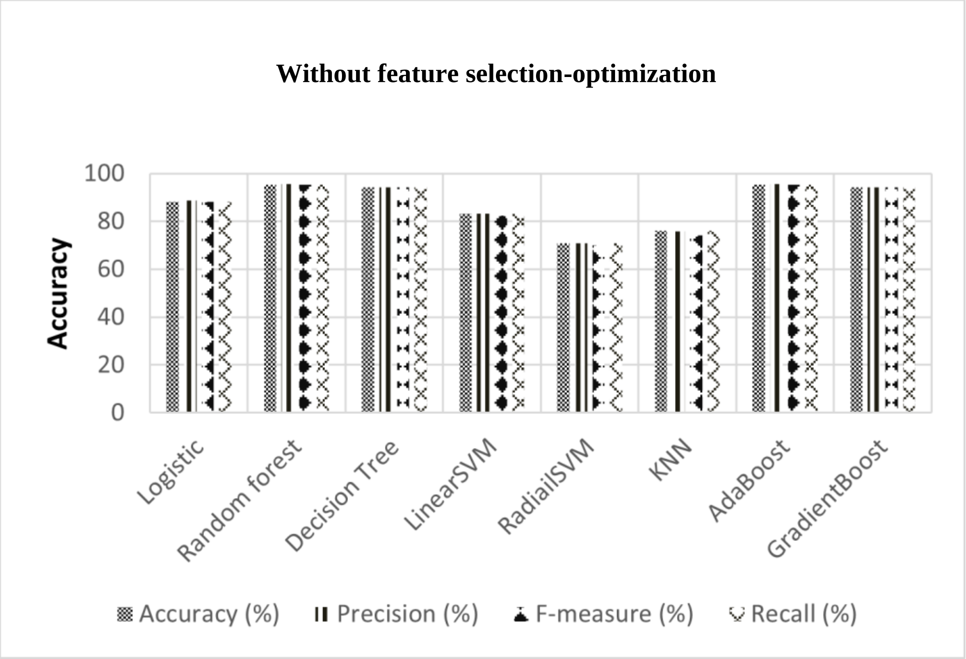


Figure 4. Classification performance of original dataset (without feature selection-optimization) using eight classifiers

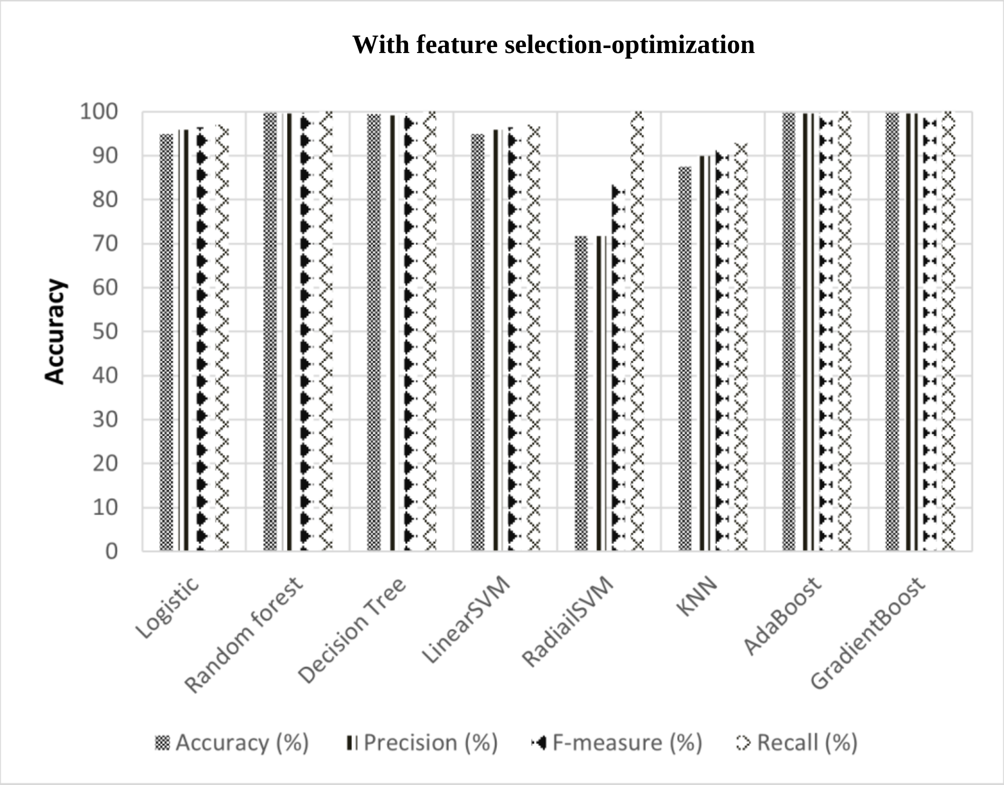


Figure 5. Classification performance of dataset with feature selection-optimization (proposed method) using eight classifiers

Figure 6 shows the comparison between the classification accuracy of original dataset (without feature selection-optimization), and classification accuracy of dataset with feature selection-optimization (proposed method). In Figure 6, the horizontal axis represents the eight classifiers that were applied to the original dataset, while the vertical axis represents the accuracy performance of these classifiers applied to the original dataset and proposed method. Figure 7 shows the confusion matrix of eight classifiers.

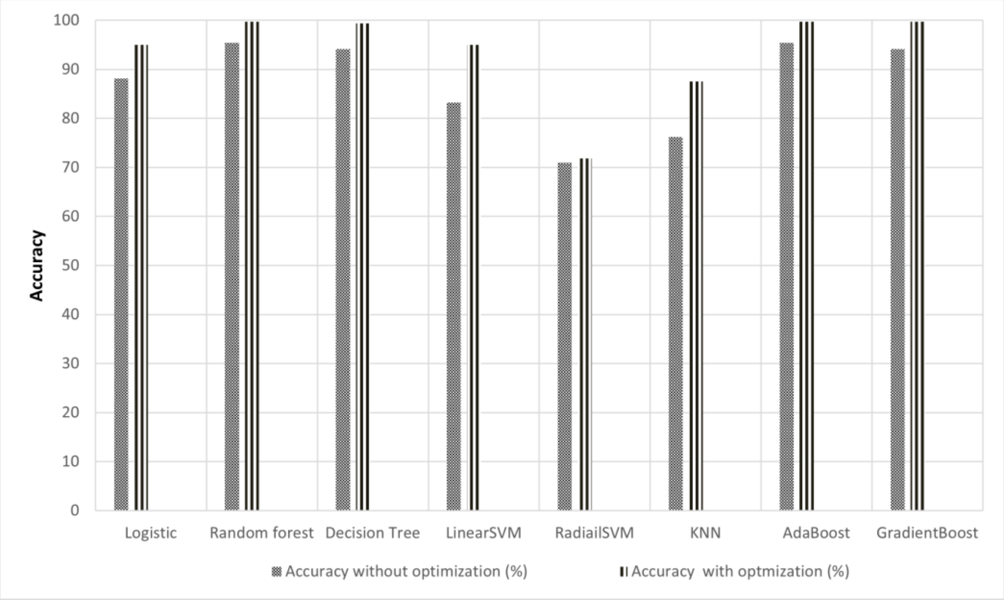


Figure 6. Comparison between classification accuracy of original dataset and proposed method

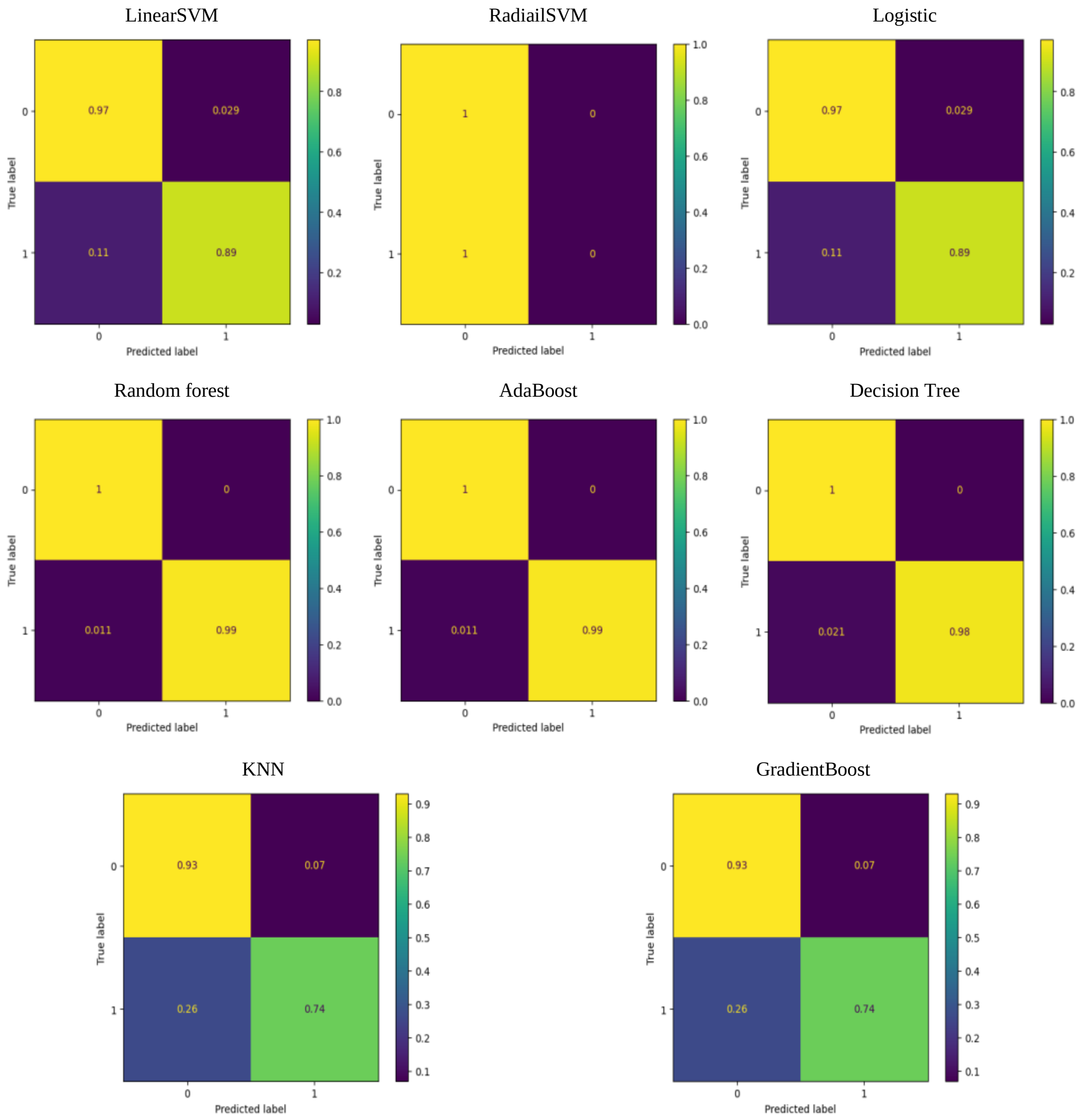


Figure 7. Confusion matrix of eight classifiers (logistic, random forest, DT, LinearSVM, RadiailSVM, KNN, AdaBoost, and GradientBoost)

5. CONCLUSION

This paper has enhanced accuracy of classification of taxi fare problem since proposed feature selection using GA optimization. The experiments applied on real-world datasets of taxi fare. The eight classifiers are employed to assess the performance of proposed technique. Several performance measures are used to evaluate performance such as accuracy, precision, F-measure, recall, TNR, and G-mean. The experimental results demonstrate that the implementation of proposed method gives higher performance than classification without optimization. Also, according to the results, accuracy and F-measure are improved since record 99.7% and precision records 99.5% evaluating by three types of classifiers (random forest, AdaBoost, and GradientBoost). Additionally, developing and evaluating hybrid classification models that incorporate various ML algorithms could improve performance. Implementing the method in real-time taxi fare prediction systems would yield valuable empirical data on its efficiency, scalability, and integration with existing technology.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Amany A. Naim	✓	✓	✓						✓					
Asmaa Hekal Omar				✓	✓					✓			✓	
Asmaa A. Ibrahim		✓				✓	✓	✓	✓					
Asmaa Mohamed	✓	✓				✓			✓	✓				
Naglaa M. Mostafa					✓				✓					✓

C	: **C**onceptualization	I	: **I**nvestigation	Vi	: **Vi**sualization
M	: **M**ethodology	R	: **R**esources	Su	: **Su**pervision
So	: **So**ftware	D	: **D**ata Curation	P	: **P**roject administration
Va	: **Va**lidation	O	: Writing - **O**riginal Draft	Fu	: **Fu**nding acquisition
Fo	: **Fo**rmal analysis	E	: Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] I. D. Lopez-Miguel, "Survey on Preprocessing Techniques for Big Data Projects," *Engineering Proceedings*, vol. 7, no. 1, p. 14, Oct. 2021, doi: 10.3390/engproc2021007014.

[2] N. Sánchez-Maroño, A. Alonso-Betanzos, and M. Tombilla-Sanromán, "Filter methods for feature selection - A comparative study," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4881, pp. 178–187, 2007, doi: 10.1007/978-3-540-77226-2_19.

[3] P. Yang, W. Liu, B. B. Zhou, S. Chawla, and A. Y. Zomaya, "Ensemble-based wrapper methods for feature selection and class imbalance learning," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7818, no. PART 1, pp. 544–555, 2013, doi: 10.1007/978-3-642-37453-1_45.

[4] A. Telikani, A. Tahmassebi, W. Banzhaf, and A. H. Gandomi, "Evolutionary Machine Learning: A Survey," *ACM Computing Surveys*, vol. 54, no. 8, pp. 1–35, Nov. 2022, doi: 10.1145/3467477.

[5] T. Kovacs, *Genetics-based machine learning*. Handbook of Natural Computing, pp. 938–986, 2012, doi: 10.1007/978-3-540-92910-9_30/FIGURES/003012.

[6] J. Zacharias, M. von Zahn, J. Chen, and O. Hinz, "Designing a feature selection method based on explainable artificial intelligence," *Electronic Markets*, vol. 32, no. 4, pp. 2159–2184, 2022, doi: 10.1007/s12525-022-00608-1.

[7] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers and Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014, doi: 10.1016/j.compeleceng.2013.11.024.

[8] J. Zhou and Z. Hua, "A correlation guided genetic algorithm and its application to feature selection," *Applied Soft Computing*, vol. 123, p. 108964, 2022, doi: 10.1016/j.asoc.2022.108964.

[9] L. A. Araújo, I. L. E. Lopes, R. M. Oliveira, S. H. G. Silva, C. S. J. E. Silva, and L. R. Gomide, "Simulated annealing in feature selection approach for modelling aboveground carbon stock at the transition between Brazilian Savanna and Atlantic Forest biomes," *Annals of Forest Research*, vol. 65, no. 1, pp. 47–63, Oct. 2022, doi: 10.15287/afr.2022.2064.

[10] A. Hashemi, M. Joodaki, N. Z. Joodaki, and M. B. Dowlatshahi, "Ant colony optimization equipped with an ensemble of heuristics through multi-criteria decision making: A case study in ensemble feature selection," *Applied Soft Computing*, vol. 124, p. 109046, Jul. 2022, doi: 10.1016/j.asoc.2022.109046.

[11] J. S. Pan, N. Liu, and S. C. Chu, "A competitive mechanism based multi-objective differential evolution algorithm and its application in feature selection," *Knowledge-Based Systems*, vol. 245, p. 108582, Jun. 2022, doi: 10.1016/j.knosys.2022.108582.

[12] S. Shanmugam and J. Preethi, "Improved feature selection and classification for rheumatoid arthritis disease using weighted decision tree approach (REACT)," *Journal of Supercomputing*, vol. 75, no. 8, pp. 5507–5519, Aug. 2019, doi: 10.1007/s11227-019-02800-1.

[13] K. Hanbay, “A new standard error based artificial bee colony algorithm and its applications in feature selection,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 4554–4567, Jul. 2022, doi: 10.1016/j.jksuci.2021.04.010.

[14] W. Xie, L. Wang, K. Yu, T. Shi, and W. Li, “Improved Multi-Layer Binary Firefly Algorithm for Optimizing Feature Selection and Classification of Microarray Data,” *SSRN Electronic Journal*, 2022, doi: 10.2139/ssrn.4073633.

[15] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The MIT Press, 1992, doi: 10.7551/mitpress/1090.001.0001.

[16] Y. Li, S. Zhang, and X. Zeng, “Research of multi-population agent genetic algorithm for feature selection,” *Expert Systems with Applications*, vol. 36, no. 9, pp. 11570–11581, Nov. 2009, doi: 10.1016/j.eswa.2009.03.032.

[17] F. Tan, X. Fu, Y. Zhang, and A. G. Bourgeois, “A genetic algorithm-based method for feature subset selection,” *Soft Computing*, vol. 12, no. 2, pp. 111–120, 2008, doi: 10.1007/s00500-007-0193-8.

[18] Z. Michalewicz, M. Schoenauer, “Evolutionary algorithms for constrained parameter optimization problems,” *Evolutionary Computation*, vol. 4, no. 1, pp. 1–32, Mar. 1996, doi: 10.1162/EVCO.1996.4.1.1.

[19] S. Katoch, S. S. Chauhan, and V. Kumar, “A review on genetic algorithm: past, present, and future,” *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021, doi: 10.1007/s11042-020-10139-6.

[20] S. D. Immanuel and U. K. Chakraborty, “Genetic Algorithm: An Approach on Optimization,” in *Proceedings of the 4th International Conference on Communication and Electronics Systems, ICCES 2019*, 2019, pp. 701–708, doi: 10.1109/ICCES45898.2019.9002372.

[21] P. K. Yadav and N. L. Prajapati, “An Overview of Genetic Algorithm and Modeling,” *International Journal of Scientific and Research Publications*, vol. 2, no. 9, pp. 1–4, 2012.

[22] K. Mehlhorn, *Data Structures and Algorithms 1*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1984, doi: 10.1007/978-3-642-69672-5.

[23] F. Weber, “Artificial Intelligence,” in *Artificial Intelligence for Business Analytics*, Wiesbaden: Springer Fachmedien Wiesbaden, 2023, pp. 33–64, doi: 10.1007/978-3-658-37599-7_2.

[24] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, “KNN model-based approach in classification,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2888, pp. 986–996, 2003, doi: 10.1007/978-3-540-39964-3_62.

[25] M. Saranya and S. Sathappan, “Survey of Crop Prediction Using Different Classification Analytical Model,” *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 20, no. 5, pp. 72–80, 2018.

[26] R. Gholami and N. Fakhari, “Support Vector Machine: Principles, Parameters, and Applications,” *Handbook of Neural Computation*, pp. 515–535, 2017, doi: 10.1016/B978-0-12-811318-9.00027-2.

[27] H. M. Sani, C. Lei, and D. Neagu, “Computational Complexity Analysis of Decision Tree Algorithms,” *Artificial Intelligence XXXV: 38th SGAI International Conference on Artificial Intelligence, AI 2018*, vol 11311, doi: 10.1007/978-3-030-04191-5_17

[28] M. Aria, C. Cuccurullo, and A. Gnasso, “A comparison among interpretative proposals for Random Forests,” *Machine Learning with Applications*, vol. 6, pp. 1–8, Dec. 2021, doi: 10.1016/j.mlwa.2021.100094.

[29] J. Pesantez-Narvaez and M. Guillen, “Weighted logistic regression to improve predictive performance in insurance,” *Advances in Intelligent Systems and Computing*, vol. 894, pp. 22–34, 2020, doi: 10.1007/978-3-030-15413-4_3.

[30] Z. A. Ali, Z. H. Abduljabbar, H. A. Tahir, A. B. Sallow, and S. M. Almufti, “eXtreme Gradient Boosting Algorithm with Machine Learning: a Review,” *Academic Journal of Nawroz University*, vol. 12, no. 2, pp. 320–334, 2023, doi: 10.25007/ajnu.v12n2a1612.

[31] M. Riansyah, S. Suwilo, and M. Zarlis, “Improved Accuracy In Data Mining Decision Tree Classification Using Adaptive Boosting (Adaboost),” *SinkrOn*, vol. 8, no. 2, pp. 617–622, 2023, doi: 10.33395/sinkron.v8i2.12055.

[32] F. Gorunescu, *Data mining: Concepts, models and techniques*, Intelligent Systems Reference Library, vol. 12. 2011, doi: 10.1007/978-3-642-19721-5.

[33] kaggle, “Dataset,” www.kaggle.com. [Online]. Available: www.kaggle.com/datasets. (Accessed: Jan. 14, 2024).

BIOGRAPHIES OF AUTHORS

Amany A. Naim received the B.Sc. degree in science, in 2009, and the M.Sc. and Ph.D. degrees in computer science, in 2015 and 2019, respectively. She is currently a lecturer in computer science with the Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, Egypt. She has published several research papers in the field of AI, machine learning, meta-heuristic optimization, data mining, and analysis. She is also a supervisor of some masters. She can be contacted at email: amany.naim@azhar.edu.eg.

Asmaa Hekal Omar received the B.Sc., M.Sc., and Ph.D. degrees from the Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, Egypt, in 2002, 2009, and 2013, respectively, where she is currently an Associate Professor in computer science. She has coauthored publications in different journals. Her research interests include machine learning and optimization. She can be contacted at email: asmaahekl@azhar.edu.eg.

Asmaa A. Ibrahim received the B.Sc. degree in science, in 2009, and the M.Sc. and Ph.D. degrees in computer science, in 2017 and 2021, respectively. She is currently a lecturer in computer science with the Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, Egypt. She has published several research papers in the field of AI, machine learning, software engineering, and internet of things. She is also a supervisor of some masters. She can be contacted at email: AsmaaabdelmoniemIbrahim1174.el@azhar.edu.eg.

Asmaa Mohamed received the B.Sc. degree in science, in 2011, the M.Sc. degree in computer science, in 2017, and the Ph.D. degree in computer science, in 2023. She is currently a lecturer in computer science at the Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, Egypt. She has published several research papers in the field of AI, machine learning, IoT, and data mining. She can be contacted at email: asmaamohamed89@azhar.edu.eg.

Naglaa M. Mostafa is a lecturer of Computer Science at Al-Azhar University (girls) in Cairo, Egypt. She was awarded a B.Sc. degree in Computer Science and pure mathematics by Faculty of Science, Menoufia University in 1996. She received her M.Sc. and Ph.D. in computer science from Faculty of Science, Al-Azhar University, Cairo, Egypt in 2009, 2012, respectively. Currently she is working at Department of Computer and Information Science, Applied College, Taibah University, Kingdom of Saudi Arabia. She can be contacted at email: NaglaaMohammed.2159@azhar.edu.eg and nmmostafa@taibahu.edu.sa.