Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 9, No. 4, December 2023, pp. 1060-1079
ISSN: 2338-3070, DOI: 10.26555/jiteki.v9i4.27014

A Hybrid Genetic Algorithm-Random Forest Regression Method for Optimum Driver Selection in Online Food Delivery

Aji Gautama Putrada[1], Nur Alamsyah[1], Ikke Dian Oktaviani[2], Mohamad Nurkamal Fauzan[1]
[1]Advanced and Creative Networks Research Center, Telkom University, Jl. Telekomunikasi No. 1, Bandung 40287, Indonesia
[2]School of Computing, Telkom University, Jl. Telekomunikasi No. 1, Bandung 40287, Indonesia

ARTICLE INFO

Article history:

Received August 23, 2023
Revised November 06, 2023
Published November 16, 2023

Keywords:

Genetic Algorithm;
Random Forest;
Optimization;
Online Food Delivery;
Driver;
Fitness Landscape

ABSTRACT

The online food delivery trend has become rapid due to the COVID-19 incident, which limited mobility, while the broader challenge in the online food delivery system is maximizing quality of service (QoS). However, studies show that driver selection and delivery time are important in customer satisfaction. The solution is our research aim, which is the selection of optimal drivers for online food delivery using random forest regression and the genetic algorithm (GA) method. Our research contribution is a novel approach to minimizing delivery time in online food delivery by combining a random forest regression model and genetic algorithms. We compare random forest regression with three other state-of-the-art regression models: linear regression, k-nearest neighbor (KNN), and adaptive boosting (AdaBoost) regression. We compare the four models with metrics including r^2, mean squared error (MSE), root mean squared error (RMSE), mean total error (MAE), and mean absolute percentage error (MAPE). We use the optimum model as the fitness function in GA. The test results show that random forest performs better than linear, KNN, and AdaBoost regression, with an r^2, RMSE, and MAE value of 0.98, 54.3, and 11, respectively. We leverage the optimum random forest regression model as the GA fitness function. The best efficiency is reducing the delivery time from 54 to 15 minutes, achieved through rigorous testing on various cases. In addition, by completing this research, we also achieve some practical implications, such as an increase in customer satisfaction, a reduction in cost, and a paramount finding in the field of data-driven decision-making. The first key finding is an optimum driver selection model in random forest regression, while the second is an optimum driver selection model in GA.

Corresponding Author:

Ikke Dian Oktaviani, School of Computing, Telkom University, Jl. Telekomunikasi No. 1, Bandung 40287, Indonesia
Email: oktavianiid@telkomuniversity.ac.id

1. INTRODUCTION

1.1. Background

Online food delivery is a food delivery service for customers that involves digital platforms such as websites or mobile apps [1]. Huq *et al.* [2] mentioned that the broader challenge in the online food delivery system is maximizing quality of service (QoS) by considering order completion properties and service expectations. On the other hand, the online food delivery trend is becoming rapid due to the COVID-19 incident, which limits mobility [3]. Several aspects are considered in online food delivery, including delivery logistics, which consists of delivery radius and delivery time estimate [4]. In addition, several studies have stated the importance of delivery time for customer satisfaction [5]. Making delivery short is important for customer satisfaction and has other positive impacts, such as reducing costs [6]. The topic of delivery time optimization in food delivery is an interesting research opportunity with all the benefits it brings.

1.2. Related Works

Optimization is paramount in online food delivery systems, and much research has been conducted in such fields. Optimization begins with designing a regression model that can relate the objective function to the factors desired to be optimized [7]. Several studies have explored the regression in delivery. Torabbeigi *et al.* [8] used a hybrid method between linear regression and mixed integer linear programming (MILP) to optimize scheduling for delivery services using drones. The goal is better drone battery savings because the battery is affected by its delivery weight. Hughes *et al.* [9] used k-nearest neighbor (KNN) regression to predict stop delivery time in logistics delivery. The test results can be used to predict whether a stop delivery time will make delivery late or not. The adaptive boosting (AdaBoost) in the research of Wang *et al.* [10] predicted hand gestures as commands in uncrewed delivery vehicles. The system can predict six different moves for different commands. Finally, the random forest was used by Errousso *et al.* [11] to predict car parking slot occupancy—that research prediction aimed to get the optimum delivery bay slot to improve the quality of delivery services. Using a regression model for optimizing driver selection is a research opportunity.

A hybrid between a regression model and the genetic algorithm (GA) method for optimization is a superior method used in various fields. Saleh *et al.* [12] used GA with multi-layer perceptron (MLP) as its fitness function to optimize the growth of cancer drugs. Yeganefar *et al.* [13] used an artificial neural network (ANN) and GA for multi-objective optimization, including surface roughness and cutting force in the milling field. The optimization parameters are cutting speed, tooth feed, depth of cut, and tool type. Torabi *et al.* [14] used ANN to predict the micro-hardness of nano-sized Cu-Cr solutions in the copper nanocomposites field. Then, GA used the ANN model as a fitness function for optimum micro-hardness. Chen *et al.* [15] used 2nd-degree polynomial regression to predict casting system structures' volume shrinkage and solidification time. Multi-objective GA is used to find the optimum value for the two metrics using the following parameters: pouring temperature, pre-heating temperature, first-part thickness ratio, second-part thickness ratio, and diameter ratio.

The identified research gap, which focuses on integrating Genetic Algorithms (GA) with an optimum regression model for driver selection in online food delivery, holds significant importance for several reasons. Firstly, traditional approaches to driver selection often rely on simplistic criteria, potentially leading to suboptimal outcomes [16]. By incorporating GA, which excels at searching for optimal solutions in complex, multi-dimensional spaces, the process becomes more sophisticated and capable of handling the diverse variables involved in driver selection. Secondly, an optimum regression model enhances the precision and accuracy of driver selection by leveraging data-driven insights.

1.3. Research Objectives and Methodology

Our research aim is to select the optimum driver for online food delivery using random forest regression and a genetic algorithm method. We then undergo several steps: obtain and explore the dataset, execute the pre-processing stage, form and design the regression models, detail the GA algorithm, run tests, and report the results. Four regression models are used: linear regression, KNN, AdaBoost, and random forest. On the other hand, we conduct the test using metrics including r^2, mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). In the GA design stage, we take into consideration the fitness function, the hyperparameters, the fitness landscape, the use of the principal component analysis (PCA) method, and lastly, the use cases.

1.4. Research Contributions and Impact

To our knowledge, a study has never used GA and regression models to optimize delivery time for online food delivery. Here are three of our scientific contributions:

1. A random forest regression model for predicting drivers in online food delivery
2. A fitness landscape of random forest regression as a fitness function in a GA solution space
3. An optimum driver selection system for online food delivery using genetic algorithms and regression analysis.

The first contribution leverages the power of ensemble learning to provide accurate and reliable driver predictions, enhancing the efficiency and effectiveness of the delivery process. The second contribution is a unique contribution that provides a deeper understanding of the interactions between the regression model's predictive capabilities and the optimization process facilitated by the GA. The last contribution addresses a critical need in the industry, which not only streamlines and automates the selection process but also ensures that it is driven by data-driven insights, leading to more efficient and effective allocation of drivers.

Our study has the potential to have a significant impact on both the online food delivery industry and the broader field of optimization:

- Impact on the Online Food Delivery Industry:
  - Enhanced Customer Satisfaction: By optimizing driver selection and reducing delivery times, the research can lead to improved customer satisfaction. Quicker deliveries are a key factor in customer experience, directly influencing user retention and loyalty.
  - Operational Efficiency and Cost Reduction: The optimized driver selection process can lead to more efficient use of resources, potentially reducing operational costs for online food delivery platforms. It can result in improved profitability and sustainability for businesses in the industry.
  - Competitive Advantage: Implementing the algorithm could provide a competitive edge for online food delivery platforms. Faster delivery times can be a significant selling point, attracting and retaining customers in a highly competitive market.
  - Adoption of Advanced Technology: The research highlights the importance of data-driven decision-making and the application of advanced algorithms in the online food delivery industry. It could serve as a precedent for integrating more sophisticated technological solutions.
- Broader Field of Optimization:
  - Innovative Approach to Hybrid Modelling: The study showcases the effectiveness of combining regression models with Genetic Algorithms (GA) for optimization. This novel approach can be applicable in various fields beyond online food delivery, offering a versatile methodology for solving complex optimization problems.
  - Insights into Fitness Landscapes: Analyzing the fitness landscape within a GA solution space provides valuable insights into the dynamics of optimization processes. This understanding can be applied in diverse domains where GA is used to search for optimal solutions.
  - Advancement in Data-Driven Decision Making: The research emphasizes the significance of utilizing data-driven approaches in optimization. This contribution can influence how optimization problems are approached and solved in various industries that rely on data-driven decision-making.

1.5. Paper Systematics

The remainder of this paper is organized according to the following systematics: Section 2, the materials and methods section, elucidates the framework for developing an optimization method tailored to online food delivery driver selection. It comprehensively outlines the algorithmic components, dataset utilization, and performance evaluation metrics employed in constructing this novel optimization approach. This section will provide a detailed explanation of the framework used to develop the optimization method for online food delivery driver selection. It will cover the specific algorithmic components involved, the utilization of the dataset, and the metrics used to evaluate the performance of the developed approach. The goal is to give a comprehensive overview of the methodology employed in constructing this novel optimization method. Section 3 discusses the results and presents the empirical outcomes of the developed optimization method for the driver selection process of online food delivery. The findings are rigorously analyzed and interpreted within the context of the research objectives, shedding light on the method's effectiveness, limitations, and potential for enhancing driver allocation strategies. This section will present the empirical outcomes of the optimization method in action for the driver selection process in online food delivery. It will provide a thorough analysis and interpretation of the findings, considering the research objectives. The section will address the method's effectiveness, discuss any identified limitations, and highlight the potential for improving driver allocation strategies in online food delivery. Lastly, Section 4 encapsulates the study's conclusion, summarizing the key insights derived from the experimentation and analysis of the proposed optimization method for online food delivery driver selection. This section reflects on the achieved outcomes and their significance and offers insights into potential avenues for further research and practical applications. The conclusion section will summarize the key insights derived from the experimentation and analysis of the proposed optimization method. It will reflect on the outcomes achieved and their significance in the broader context of online food delivery. Additionally, this section will offer insights into potential future avenues for research and practical applications, suggesting areas where further advancements or implementations could be pursued based on the study's findings.

2. METHODS

We carry a methodology to achieve our research aim. First, we obtained the online food delivery dataset from Kaggle. We then pre-processed the dataset with label encoding. After that, we form a regression model based on the dataset. We compare four regression models: linear regression, KNN, AdaBoost, and random forest. We compare the four models with metrics including r^2, MSE, RMSE, MAE, and MAPE. We use the

optimum model as the fitness function in GA. We compare several cases to see how our model optimizes delivery time. Fig. 1 illustrates our proposed method in the form of a flow chart.

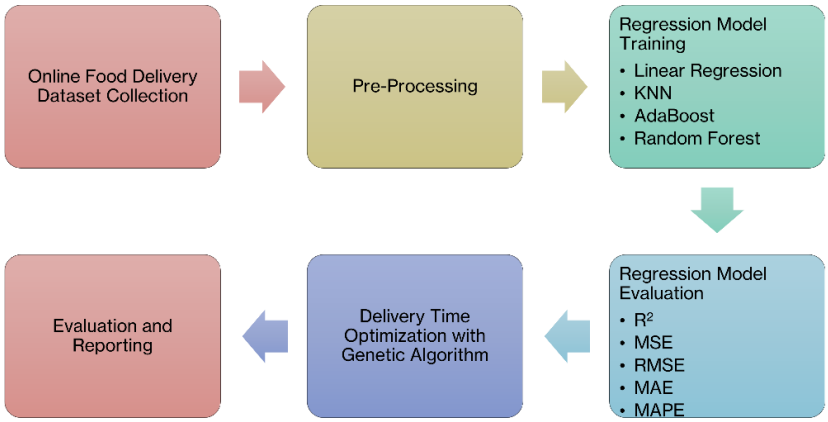

Fig. 1. Our Proposed Methodology

2.1. Online Food Delivery Dataset Collection

Online food delivery is a system that utilizes digital platforms and involves mobile apps or websites. We achieved our online food delivery dataset from Kaggle and uploaded by Bhanuratap Biswas *et al.* The dataset contains India's online food delivery dataset, with 45,594 data items and 11 variables. Following are the names and explanations for each variable:

1. "ID": The identity of the online food delivery. The format is string.
2. "Delivery_person_ID": The identity of the driver. There are 1320 unique driver IDs in the dataset. The format is string.
3. "Delivery_person_Age": The age of the driver. The age range of drivers in the dataset is 15 to 50. The format is integer.
4. "Delivery_person_Ratings": The rating received by the driver in delivery. The range is 0 to 6, then the format is float.
5. "Restaurant_latitude": The latitude position of the restaurant. The format is float.
6. "Restaurant_longitude": The longitude position of the restaurant. The format is float.
7. "Delivery_location_latitude": The latitude position of the delivery location. The format is float.
8. "Delivery_location_longitude": The longitude position of the delivery location. The format is float.
9. "Type_of_order": The food ordered in the online food delivery system. There are four order types: "Snack," "Drinks," "Buffet," and "Meal." The format is string.
10. "Type_of_vehicle": The driver's vehicle type. There are four vehicle types: "motorcycle," "scooter," "electric_scooter," and "bicycle." The format is string.
11. "Time_taken(min)": Time is taken for delivery in minutes. The range is from 10 to 54 minutes. The format is integer.

The longitude and latitude data related to restaurant locations or destinations are usually taken using a GPS module [17]. Our regression model uses "Delivery_person_ID" as the dependent variable and ten other features as independent variables.

2.2. Pre-Processing

We also use several pre-processing methods before the regression model training phase. First, we use the label encoder to change the variable from a string to a representative integer [18]. Then, we deal with missing values by removing rows with "not a number" (NaN) values. In addition, we use the Pearson correlation to observe correlations between the independent and dependent variables.

Label encoding is a technique that transforms categorical data into numerical values, making it compatible with machine learning algorithms that require numerical input [19]. In implementing the *label encoding*, each category in a categorical feature is assigned a unique integer label. For example, if we have a categorical feature like " Type_of_order" with labels: "Snack," "Drinks," "Buffet," and "Meal," label encoding would assign numerical values like 0, 1, 2, and 3, respectively. While label encoding facilitates the inclusion of categorical data in machine learning models, it is important to note that it introduces ordinal relationships that may not exist in the original data. For instance, the numerical values imply an ordering that may not be meaningful in

the context of the categorical feature. Therefore, label encoding is most suitable for nominal categorical variables with no inherent order among the categories.

Removing data items with missing values, often accomplished using dropna in pandas, is a crucial step in data pre-processing [20]. This process involves identifying and eliminating rows or columns that contain incomplete or NaN (Not a Number) values. The impact on the dataset is twofold. Firstly, it reduces potential noise or inaccuracies in the analysis, as missing data can lead to biased results or erroneous conclusions. Secondly, it streamlines the dataset, making it more manageable and suitable for further processing, such as statistical modeling or machine learning. However, it is essential to exercise caution and carefully consider which data items to remove, as excessive removal of rows or columns could lead to a loss of valuable information. Therefore, this step requires a balanced approach, where researchers weigh the benefits of cleaner data against the potential loss of information.

Implementing Pearson correlation involves calculating the statistical measure of association between pairs of numerical variables in a dataset [21]. This technique quantifies the linear relationship between variables, providing insights into their mutual dependencies. By computing the Pearson correlation coefficient, which ranges from -1 to +1, a value closer to +1 indicates a strong positive correlation, while a value closer to -1 signifies a strong negative correlation. A correlation of 0 indicates no linear relationship. The impact of applying Pearson correlation to a dataset is profound, as it allows researchers to identify significant associations between variables. This information is crucial for tasks like feature selection, where it helps in understanding which features have a meaningful impact on the target variable.

Additionally, it aids in identifying potential multicollinearity issues in regression analysis, providing insights for more accurate modeling. Overall, implementing the Pearson correlation empowers researchers to decide which variables to include or exclude in their analyses, leading to more robust and accurate statistical models. The formula for the Pearson correlation (r) is as (1).

$$r = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2(b_i - \bar{b})^2}} \tag{1}$$

where a is the first variable, b is the second variable, $\bar{a}$ is the average of the first variable, $\bar{b}$ is the average of the second variable, and n is the dataset size.

2.3. Random Forest Regression Model

Random forest is a highly versatile and powerful regression technique known for its robustness and accuracy. One of its key strengths lies in its ability to handle large and complex datasets with many features, making it well-suited for a wide range of real-world applications. Additionally, random forest can naturally handle both categorical and numerical variables without the need for extensive data pre-processing, which simplifies the modeling process. This versatility makes random forest an attractive choice for regression tasks where the relationships between variables may be intricate or not easily captured by simpler models.

Random forest is a part of the ensemble learning method that makes several weak decision trees and combines the predicted results of each tree [22]. It is an ensemble learning type of bootstrap and aggregating (bagging), where bootstrap is a random sampling process for each decision tree, and aggregating is the process of combining the results of all trees [23]. In random forest regression, aggregation is done by averaging the results of all weak learners [24]. Fig. 2 shows the algorithm of random forest regression for training and prediction. X is the independent variable for training, y is the dependent variable for training, T is the independent variable for prediction, W is the number of weak learners, and p is the final prediction of random forest regression. Then, “Bootstrap” is a function that calls the bootstrap sampling process. In addition, “Train” is a function that calls the training process for weak learners. Finally, F is a model for weak learners.

Furthermore, random forest provides a built-in mechanism for feature importance assessment. By evaluating the impact of each variable on the model's predictive performance, researchers can gain valuable insights into which features are most influential in making accurate predictions. This information can be crucial for understanding the underlying dynamics of the dataset and can guide future analyses or interventions. Additionally, the ensemble nature of random forest reduces the risk of overfitting, as it combines the predictions of multiple weak learners. This results in a more stable and reliable regression model less prone to capturing noise or idiosyncrasies in the data. Overall, random forest regression offers a robust and flexible approach that excels in capturing complex relationships and delivering accurate predictions across a wide range of regression tasks.

Algorithm 1: Random Forest Regression

Data: X, y, T, W
Result: p
for $i \in W$ **do**
  /* Bootstrap Sampling */
  $X_i, y_i \leftarrow Bootstrap(X, y)$;
  /* Tree Construction */
  $F_i \leftarrow Train(X_i, y_i)$;
end
/* Prediction */
for $i \in W$ **do**
  $p_i \leftarrow F_i(T)$;
end
/* Equation for prediction aggregation of weak learners */

$$p = \frac{1}{W} \sum_{i=0}^{W} p_i \quad (2)$$

Fig. 2. Pseudocode of the random forest training and prediction algorithm

2.4. Benchmark Regression Models

Using benchmark models is a crucial step in evaluating the performance of a proposed random forest regression model. Here is our benchmarking methodology:

1. Select Benchmark Models:
   Choose a set of benchmark models that are relevant to the regression task. These models should represent different approaches to solving the same problem. Common benchmarks include simpler models like linear regression, KNN, or AdaBoost regression.
2. Split the Dataset:
   We divide our dataset into training and testing sets. The training set is used to train both the benchmark models and the proposed random forest regression model. The testing set is reserved for evaluating their performance.
3. Train Benchmark Models:
   We train each of the selected benchmark models using the training data. We ensure that the same training data is used for all models to ensure a fair comparison.
4. Evaluate Benchmark Models:
   We use the testing set to evaluate the performance of each benchmark model. We calculate relevant regression evaluation metrics such as R-squared, MAE, MSE, RMSE, and MAPE.
5. Train the Random Forest Model:
   Train the proposed random forest regression model using the same training data used for the benchmark models.
6. Evaluate the Random Forest Model:
   We use the same testing set to evaluate the performance of the random forest regression model. Calculate the same set of evaluation metrics used for the benchmark models.
7. Compare Performance:
   Compare the performance metrics of the random forest model against those of the benchmark models. This comparison provides insights into whether the random forest model outperforms or is comparable to the benchmark models.
8. Consider Trade-offs:
   We consider computational complexity, interpretability, and model assumptions when interpreting the results. Some benchmark models may have advantages in specific areas even if they perform slightly worse in terms of predictive accuracy.
9. Iterate and Refine:
   Based on the benchmarking results, we refine the random forest model. It could involve fine-tuning hyperparameters, feature engineering, or exploring different variations of the model architecture.

The objective we desire by systematically comparing the performance of the proposed random forest regression model against benchmark models is to gain a clear understanding of its relative strengths and weaknesses, helping us make informed decisions about its suitability for our specific regression task.

We benchmarked random forest regression against three other state-of-the-art regression models: linear regression, KNN, and AdaBoost Regression. Linear regression maps independent and dependent variables with a linear function [25]. The best fit of linear regression is when the smallest sum of squared differences between the actual dependent variable and the predicted dependent variable is obtained [26]. The reason we use linear regression is because one of the state-of-the-art researchers, Torabbeigi *et al.* [8], used linear regression to optimize scheduling for delivery services using drones. Parameter tuning in linear regression involves optimizing the model's hyperparameters to achieve the best possible performance. One essential step is selecting the appropriate regularization technique, which helps prevent overfitting and improves generalization. It includes choosing between L1 (Lasso) and L2 (Ridge) regularization, each introducing a penalty term on the coefficients to control their magnitude. The next step is determining the strength of regularization through the alpha parameter. A higher alpha value increases the strength of the penalty, potentially leading to more coefficients being pushed towards zero. Cross-validation is crucial for evaluating different combinations of hyperparameters, as it provides a reliable estimate of model performance. Grid search or random search techniques can be employed to explore the hyperparameter space and find the best combination systematically. Finally, once optimal hyperparameters are identified, they can be used to train the final linear regression model on the entire dataset for deployment or further evaluation.

KNN is a machine learning that makes predictions based on the shortest distance of a dependent variable to the dependent variable in feature space [27]. As the KNN classification does majority voting to decide on the final class, the final result of the counterpart KNN regression prediction is determined by averaging all ‘k’ neighbors with the closest distance to the independent variable from the predicted data [28]. The reason we use KNN regression is because one of the state-of-the-art research, Hughes *et al.* [9], used KNN regression to predict stop delivery time in logistics delivery. The test results can be used to predict whether a stop delivery time will make delivery late or not. Parameter tuning for KNN regression involves several crucial steps to optimize model performance. Firstly, selecting the appropriate value of 'k' - the number of nearest neighbors to consider - is pivotal. A small 'k' may lead to overfitting, while a large 'k' might lead to underfitting. Cross-validation techniques, such as k-fold cross-validation, help assess model performance across different 'k' values and choose the one that yields the best results.

Additionally, the choice of distance metric, such as Euclidean or Manhattan distance, can significantly impact the performance of the KNN regression model. Experimenting with different distance metrics and assessing their effects on model accuracy is an essential tuning step. Furthermore, feature scaling, such as standardization or normalization, can influence the distance calculations and should be considered during parameter tuning. Lastly, it is important to evaluate the impact of any additional parameters specific to the KNN algorithm being used, such as the type of weighting scheme (e.g., uniform or distance-weighted) when aggregating predictions from neighbors. By systematically adjusting these parameters and assessing their impact on model performance through cross-validation, one can fine-tune a KNN regression model to achieve optimal results for a specific dataset.

AdaBoost regression is also an ensemble learning with an average aggregation of all weak learners, like the random forest, but it is one of the boosting types [29]. In the boosting process on AdaBoost, weak learners are arranged serially, where the next weak learner is an improvement from the previous weak learner by increasing the weight of incorrectly predicted data [30]. The reason we use KNN regression is that one of the state-of-the-art research, Wang *et al.* [10], predicted hand gestures as commands in uncrewed delivery vehicles using AdaBoost regression. Parameter tuning for AdaBoost regression involves several key steps to enhance model performance. Initially, selecting the base estimator is crucial, as it determines the type of weak learner used in the ensemble. Common choices include decision trees or regression trees. Experimenting with different base estimator types and assessing their impact on model accuracy is an important tuning step.

Additionally, determining the number of weak learners (n_estimators) in the ensemble is pivotal. While a higher number of estimators can lead to improved performance, it also increases computational complexity. Cross-validation techniques, such as grid search or randomized search, can help identify the optimal number of estimators. Adjusting the learning rate is another crucial step. A lower learning rate reduces the contribution of each weak learner, potentially leading to better generalization. It is important to balance the learning rate and the number of estimators. Finally, assessing the impact of additional parameters specific to the base estimator (e.g., maximum depth of trees) is essential for fine-tuning the AdaBoost regression model. By systematically adjusting these parameters and evaluating their impact on model performance through cross-validation, one can optimize an AdaBoost regression model for a specific dataset.

2.5. Performance Metrics

We use r^2, MSE, RMSE, MAE, and MAPE to compare the four models. The r^2 metric is the ratio between explained variance and total variance, where the greater the ratio, the better the model [31]. The r^2 metric provides a valuable measure of how well the independent variables in a regression model explain the variation in the dependent variable. Specifically, it quantifies the proportion of the total variance in the dependent variable that can be accounted for by the independent variables included in the model. In other words, it assesses the goodness of fit of the model. An r^2 value closer to 1 indicates that a larger proportion of the variance is explained, implying a better fit.

Conversely, an r^2 value closer to 0 suggests that the model is less effective at explaining the variability in the data. It is important to note that while a high r^2 value is desirable, it does not necessarily imply causation, and caution should be exercised in making causal inferences solely based on this metric. Additionally, it is advisable to consider other evaluation metrics and conduct a thorough analysis of the model's assumptions and performance in conjunction with r^2 to ensure a comprehensive assessment of the regression model's effectiveness.

Using quadratic operations, MSE emphasizes large rather than small errors [32]. MSE is a widely used metric in regression analysis that measures the average squared difference between the observed and predicted values in a dataset. By squaring the differences, MSE emphasizes larger errors, making it particularly sensitive to outliers or significant deviations from the predicted values. This characteristic of MSE is especially valuable in scenarios where accurately capturing and minimizing large discrepancies between observed and predicted values is paramount. However, it is important to note that because MSE involves squaring the errors, it can yield larger values than other metrics, potentially making interpretation less intuitive. Therefore, while MSE is a valuable tool for assessing the overall performance of a regression model, we also use additional evaluation metrics and perform a thorough analysis of the model's behavior to gain a comprehensive understanding of its effectiveness.

The root operation in RMSE makes RMSE results more interpretable than MSE [33]. RMSE is a widely used metric in regression analysis that builds upon the MSE by taking the square root of the average squared differences between observed and predicted values. This transformation to RMSE makes the results more interpretable and aligns them with the original scale of the dependent variable. Unlike MSE, which yields values in squared units, RMSE provides results in the same units as the dependent variable, making it easier to grasp the magnitude of prediction errors. It is particularly valuable in practical applications where having a clear understanding of the scale of errors is essential for decision-making. However, it is important to remember that RMSE, like MSE, is sensitive to outliers and larger errors, so we complement its interpretation with a thorough examination of the model's behavior and consideration of other relevant evaluation metrics.

Replacing the squared operation with absolute in MAE makes MAE more robust against outliers [34]. MAE is a key metric used in regression analysis to quantify the average absolute differences between observed and predicted values in a dataset. Unlike MSE and RMSE, which square the errors, MAE takes the absolute value of the differences, effectively treating all discrepancies equally regardless of their magnitude. This characteristic makes MAE more robust against outliers and large deviations in the data. By emphasizing the absolute differences, MAE provides a more straightforward and intuitive measure of prediction accuracy. It is particularly beneficial in situations where the impact of outliers needs to be minimized or where a more balanced assessment of the model's performance is desired. While MAE may not penalize large errors as heavily as MSE or RMSE, it offers a valuable alternative for evaluating the accuracy of a regression model, providing a clear and interpretable measure of the average prediction error.

The percentage operation in MAPE makes this metric good for measuring errors with a wide range of variations [35]. MAPE is a widely used metric in forecasting and regression analysis that quantifies the average percentage difference between observed and predicted values. This metric is particularly valuable in scenarios where understanding the relative magnitude of errors is crucial. By expressing errors as percentages of the observed values, MAPE provides a normalized measure that is independent of the scale of the data. It makes MAPE suitable for evaluating models across different datasets or variables, allowing for a more comprehensive assessment of forecasting accuracy.

Additionally, MAPE's percentage-based calculation makes it robust against outliers and variations in data magnitude. It is particularly useful when dealing with datasets with a wide range of values or comparing models across diverse contexts. However, it is important to note that MAPE can be sensitive to cases where the observed values are close to zero, potentially leading to large percentage errors. In such cases, additional caution and consideration of alternative metrics may be warranted.

We also use K-fold cross-validation, a good method to check for overfitting in a regression model [36]. We use K = 5, where based on the number of K, the dataset is divided into five parts, then four parts are used

for training, while one part is used for testing (with r^2), then a round-robin iterates the same operation for all sections [37]. K-fold cross-validation is a robust and widely employed technique in machine learning for assessing the performance and generalizability of a regression model. By partitioning the dataset into K subsets, where K is typically set to a value like 5 or 10, the model is trained and evaluated multiple times. During each iteration, K-1 of the subsets are utilized for training, while the remaining subset is employed for testing. This process is performed round-robin, ensuring that each subset is used as the testing set exactly once. The results from each fold are then averaged to provide a comprehensive evaluation of the model's performance. This approach is particularly effective in detecting overfitting, as it systematically tests the model's ability to generalize to unseen data. Using K-fold cross-validation, researchers can gain valuable insights into how well the model will likely perform on new, unseen data, providing a robust assessment of its predictive capabilities.

2.6. Genetic Algorithm Optimization

GA is a meta-heuristic optimization inspired by evolution and natural selection [38]. This method solves complex problems using the iterative generation of measurable solutions [39]. Fig. 3 shows the pseudocode of GA. P describes the number of population generated in each generation. G describes the number of iterations for a generation. I is a generated individual. Several processes and terms need to be known in GA. The fitness function is a function to measure the quality of an individual in GA [40]. Crossover is the process of combining the parameters of two individuals and combining them into a new individual [41]. Mutation is the process of changing the parameters of an individual [42]. The generation process can be completed if convergence has been reached before the number of generations, called the convergence criterion [43]. The generation process of the parameters depends on the range of values for each specified parameter. The number of parameters for each individual depends on the number determined at the beginning. The number of selections, crossovers, and mutations is based on the initial setting.

Algorithm 2: Genetic Algorithm
Data: P, G
Result: I
`/* Initialization */`
Randomly generate I as much as P;
$i \leftarrow 0$;
while $i < G$ **do**
  `/* Fitness Evaluation */`
  Evaluate the fitness of each I;
  `/* Selection */`
  Select I with high fitness values for parents of the new generation;
  `/* Crossover */`
  Perform crossover by combining parameters from parents for their offspring;
  `/* Mutation */`
  Perform mutation by changing the parameters of some offspring;
  `/* New Generation */`
  Create new Is consisting of parents, crossovers, and mutation;
  $i \leftarrow i + 1$;
end

Fig. 3. Pseudocode of the GA

Defining the fitness function for a Genetic Algorithm (GA) involves quantifying how well a particular solution (or individual) within the population performs concerning the problem at hand [44]. The fitness function acts as a metric to evaluate the quality of each solution. Some steps are needed to define a fitness function for a GA. The goal of the optimization problem needs to be clearly defined. In our research, the goal of the optimization problem is to minimize the delivery time in the online food delivery system. In the next step, we must translate the objective into a metric. In our case, this could be expressed as "minimizing delivery time in minutes." Subsequently, for each candidate solution (individual), we must apply the metric to quantify its performance. This step requires calculating the metric value based on the solution's characteristics or parameters. It is where the optimum regression model takes the hand. In addition, we must ensure that the metric is appropriately scaled. We are aiming to minimize delivery time, and then we must make sure that lower values represent good solutions. If there are constraints in the problem, incorporate them into the fitness function. In our research, the constraints are the other features, such as the location of the restaurant, the location of the customer, and the driver. Lastly, we must test the fitness function with sample solutions to

ensure that it accurately reflects the problem's optimization goal. Here, we use four different use cases to test the model. Fig. 4 shows the process in the form of a flowchart.

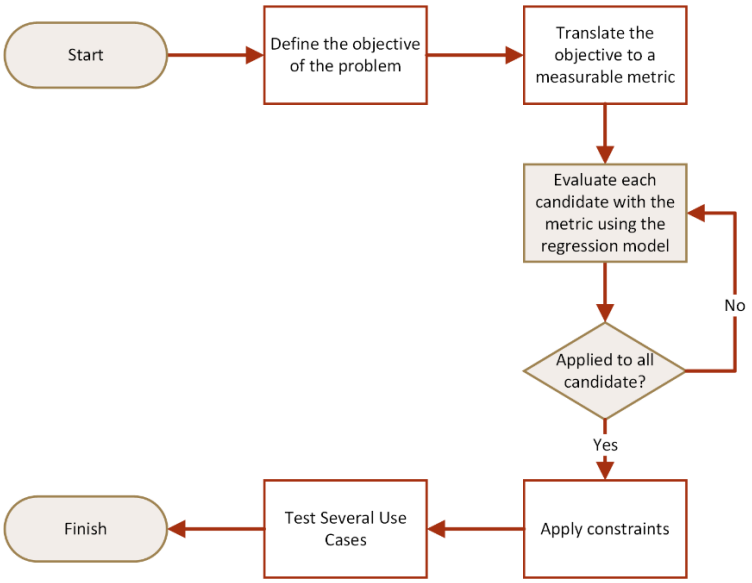

Fig. 4. Defining a fitness function and completing the GA optimization

3. RESULTS AND DISCUSSION

3.1. Results

There are four variables with data type strings in the dataset: "ID," "Delivery_person_ID," "Type_of_order," and "Type_of_vehicle." We use the label encoder to convert the four variables into integers. We found six data items containing NaN. We discarded the six data items and used the rest. We analyzed the dataset using the Pearson correlation. Fig. 5 shows the Pearson correlation calculated matrix. We use "Delivery_person_ID" as the dependent variable to evaluate the relationship between this variable and other variables. The three variables with the strongest correlation with "Delivery_person_ID" are "ID," "Restaurant_longitude," and "Delivery_location_longitude," with values of -0.2, -0.2, and 0.2, respectively.

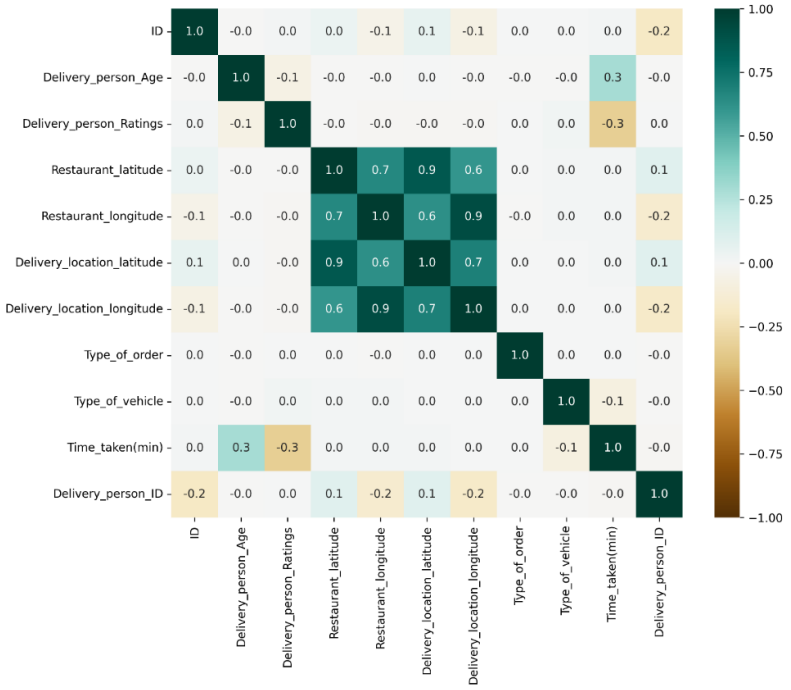

Fig. 5. Pearson correlation of food delivery features

We conducted training on four regression models using an online food delivery dataset. We use 50% dataset for training and 50% dataset for testing. The compared models are linear regression, KNN, AdaBoost, and Random Forest. Fig. 6 shows the results of the performance comparison of the four models. The r^2 metric provides valuable insights into how well the models explain the variance in the data. The Random Forest model demonstrates an impressive r^2 value of 0.98, indicating that it can explain a substantial proportion of the variance in the dependent variable. It suggests a strong and reliable predictive performance. The AdaBoost, KNN, and Linear Regression models also exhibit respectable r^2 values, albeit slightly lower than that of the Random Forest. It indicates that they can still capture a significant portion of the variance in the data, albeit to a slightly lesser extent.

Using CV scores is a critical step in assessing potential overfitting in the models. All four models showcase CV scores that closely approximate their respective r^2 values. This alignment indicates that the models are not exhibiting signs of overfitting, as their performance remains consistent when tested on different subsets of the data. This robustness is a positive sign, suggesting that the models will likely generalize well to new, unseen data.

Examining metrics such as MSE, Root Mean Squared Error RMSE, Mean Absolute Error MAE, and Mean Absolute Percentage Error MAPE provides additional granularity in evaluating model performance. The Random Forest model demonstrates the lowest values across all these metrics compared to the other three models. It indicates that the Random Forest model produces predictions that are, on average, closest to the actual values in the dataset. Additionally, the presence of differing RMSE and MAE values suggests that the RMSE metric is sensitive to outliers in the dataset. The Random Forest model may be particularly effective at capturing and accounting for data points that deviate significantly from the overall trend.

In summary, the Random Forest model consistently outperforms the other models across various evaluation metrics, including r^2, MSE, RMSE, MAE, and MAPE. It suggests it is the most suitable model for accurately predicting the dependent variable in this context. The other models also exhibit respectable performance, but they may benefit from further refinement or consideration of additional features to improve their predictive capabilities.

Regression plots serve as a crucial visual diagnostic tool for evaluating the performance and goodness of fit of regression models. In Fig. 7, the scatter plot between actual and predicted values for each of the four regression models provides valuable insights into their predictive capabilities. The linear regression line acts as a reference, representing the ideal scenario where predicted values perfectly align with actual values. By examining the dispersion of data points around this line, we gain a direct visual assessment of how well the models can capture the underlying relationships in the data.

Upon careful inspection, it is evident that the linear and KNN regression models exhibit a more scattered distribution of data points around the regression line. It suggests that these models struggle to accurately predict the dependent variable, as evidenced by the increased variability in their predictions. On the other hand, the AdaBoost regression model demonstrates a more concentrated clustering of data points around the regression line, indicating a higher degree of precision in its predictions compared to linear regression and KNN regression. AdaBoost can capture more of the underlying structure in the data.

Furthermore, the regression plot for the random forest regression model showcases an even tighter clustering of data points around the regression line, indicating a higher level of accuracy and precision in its predictions compared to all other models. This visual observation aligns with the quantitative assessments provided by the evaluation metrics. Overall, the regression plots offer a powerful visual confirmation of the model performance, reaffirming the quantitative findings and providing additional confidence in the suitability of the random forest regression model for this specific regression task.

Because random forest regression performs best, we use the model as a fitness function for the next step: optimization with the genetic algorithm. A fitness landscape can explain changes in the fitness function to the two input variables of the regression model. So that we can do a fitness landscape in the form of a 3D plot, we first transform the independent variables from the online food delivery dataset into two dimensions using principal component analysis (PCA). Fig. 8 shows the fitness landscape of the random forest regression as the solution space of the GA. The first attribute that can be observed is the surface. The surface shows complexity because several local optima are monitored. The plot also shows valleys and peaks. Because our optimum definition is the lowest delivery time, the optimum region is shown by principal component 1 > -10000 and principal component 2 > 60. The need for a trade-off is seen when principal component 1 must be small while principal component 2 must be large. Nevertheless, the fitness will rise again when the principal component 1 is too small.

Before implementing GA, we did parameter tuning first. Parameter tuning in GA is done to get GA with optimal performance and effective optimization. Parameter tuning in GA can involve trial and error,

experiments, and a deep understanding process. The point is to get a balance between exploration and exploitation. Table 1 shows the results of our GA parameter tuning stage. We are tuning five parameters: generation, population, mutation probability, crossover probability, and parent portion. We give four candidate values for each parameter, so there are five tests with four iterations. The best values for each parameter are 100, 10, 0.3, 0.3, and 0.3, respectively.

We retrieve the four worst delivery cases in the dataset based on the "Time_taken(min)" variable. We hypothesize that using GA, food delivery from restaurants to delivery locations in the four cases can be optimized because of better driver choices. We run the GA with the parameter tuning results from the previous test. Fig. 9 shows the convergence curve of the GA in the four cases. The beginning of the generation can show different values because the GA starts from a different beginning and explores the fitness function. Generations zero through 20 show a significant decline. The decline shows exploration or when GA gets a promising region in the solution space. The 20th generation to the end shows a stable value and little variation. That part is the convergence phase. A plateau is not observable in the convergence curve. This result of observation shows that GA is not trapped in the local optimum. Because our goal is to minimize, the most successful case is case 3. Case 3 shows the lowest objective function and also the largest decline.

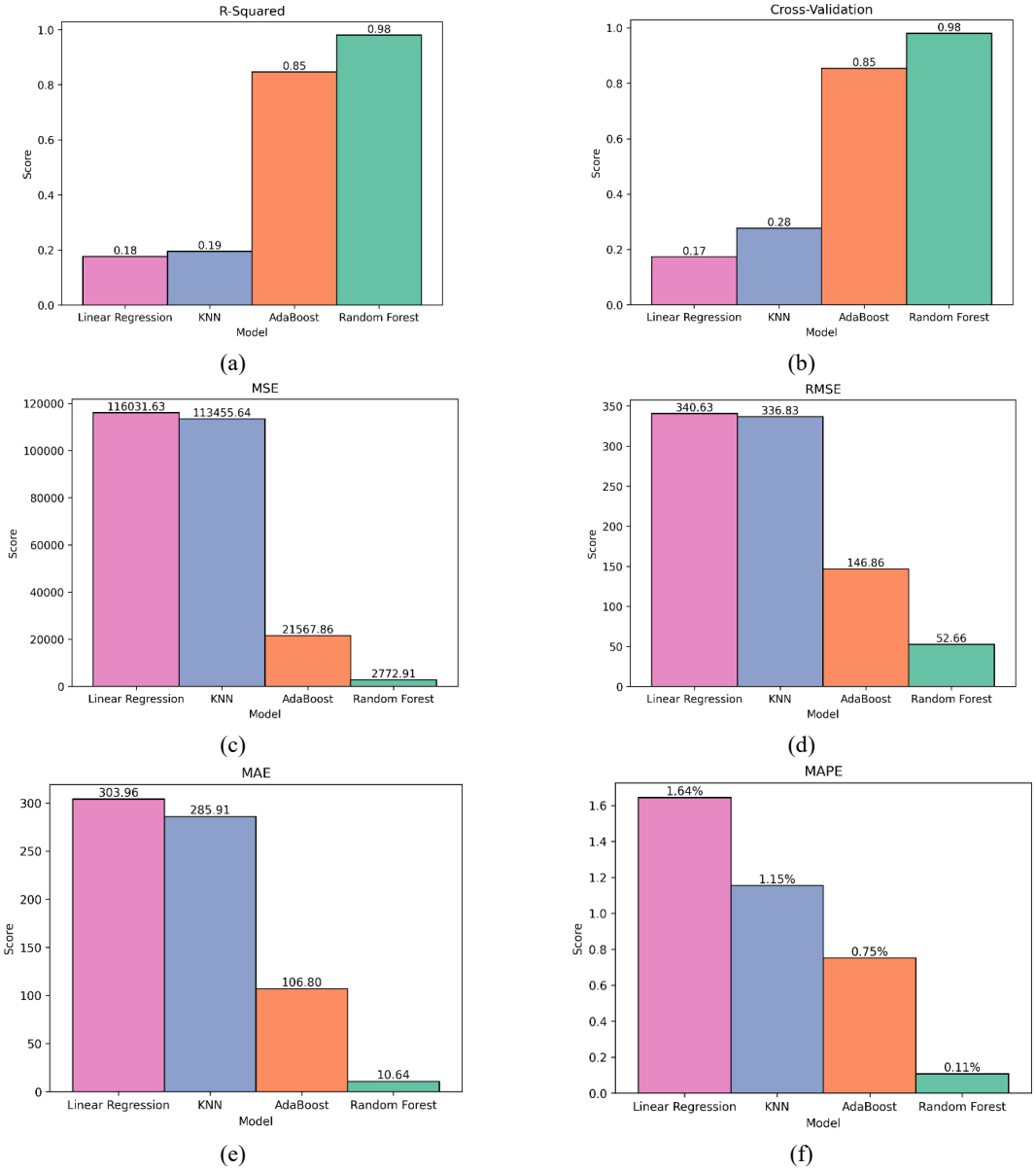

Fig. 6. Performance Comparison of Regression Models on Predicting Online Food Delivery Time: (a) R-squared (b) Cross-validation (c) MSE (d) RMSE (e) MAE (f) MAPE

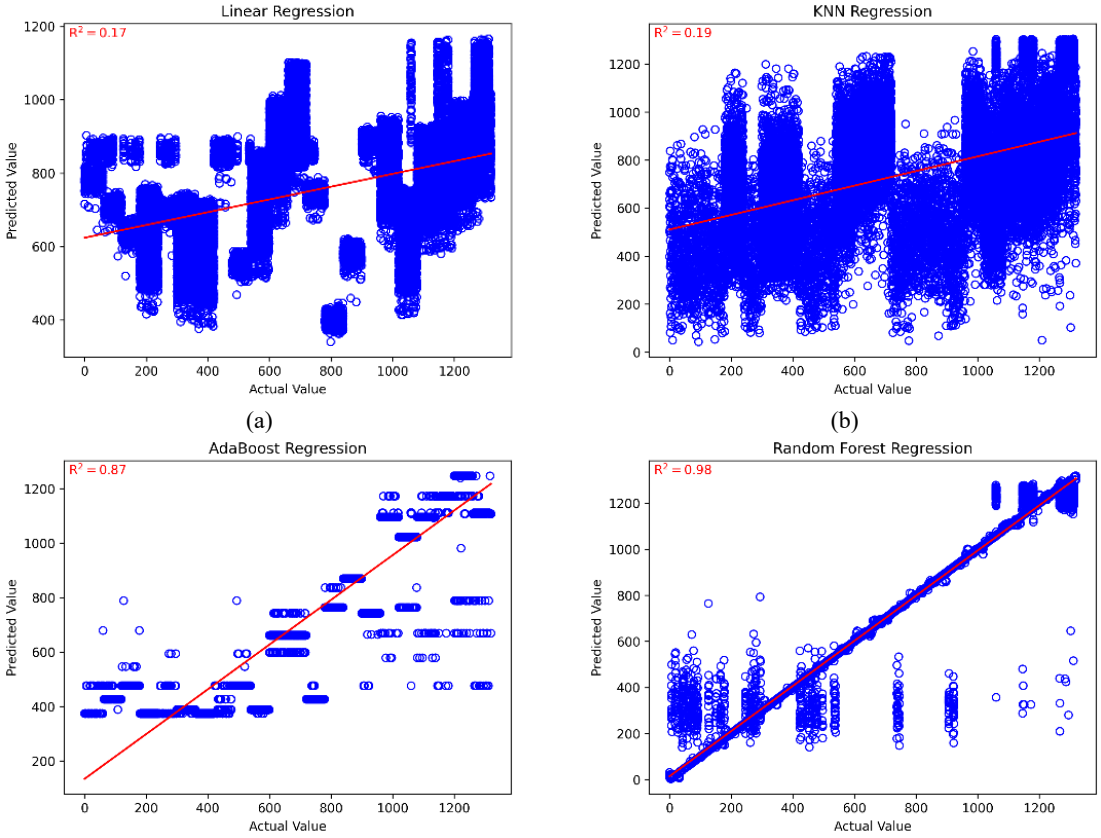

Fig. 7. Regression plots of four regression models on predicting online food delivery time: (a) Linear regession (b) KNN regression (c) AdaBoost regression (d) Random forest regression

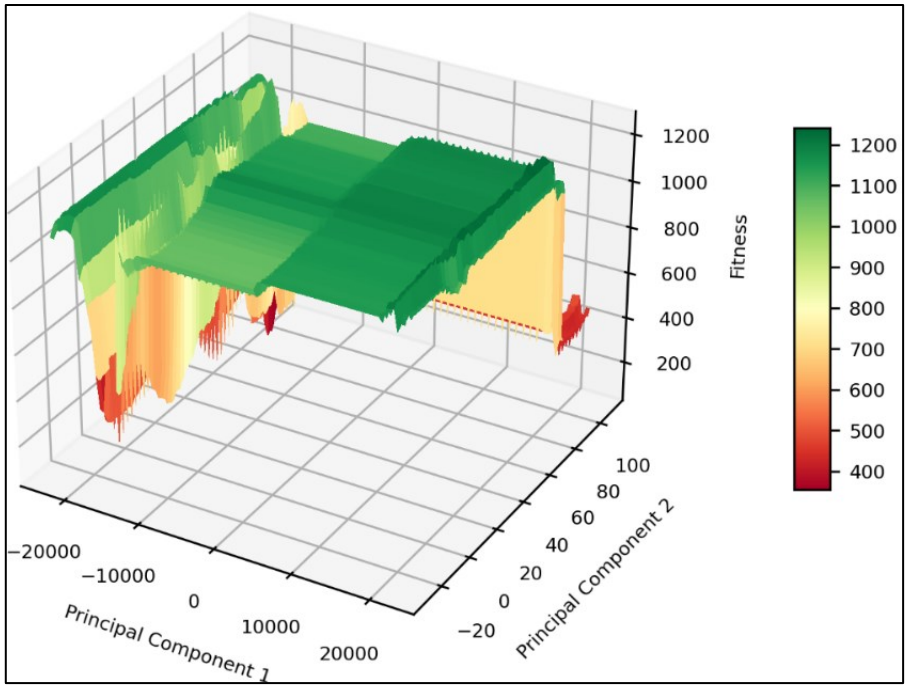

Fig. 8. The fitness landsacpe of the random forest fitness function in the GA solution space

Table 1. GA Parameter Tuning

No.	Parameter Name	Parameter Values	Best Value
1	Generation	100, 200, 300, 400	100
2	Population	10, 20, 30, 40	10
3	Mutation Probability	0.3, 0.4, 0.5, 0.6	0.3
4	Crossover Probability	0.3, 0.4, 0.5, 0.6	0.3
5	Parents Portion	0.3, 0.4, 0.5, 0.6	0.3

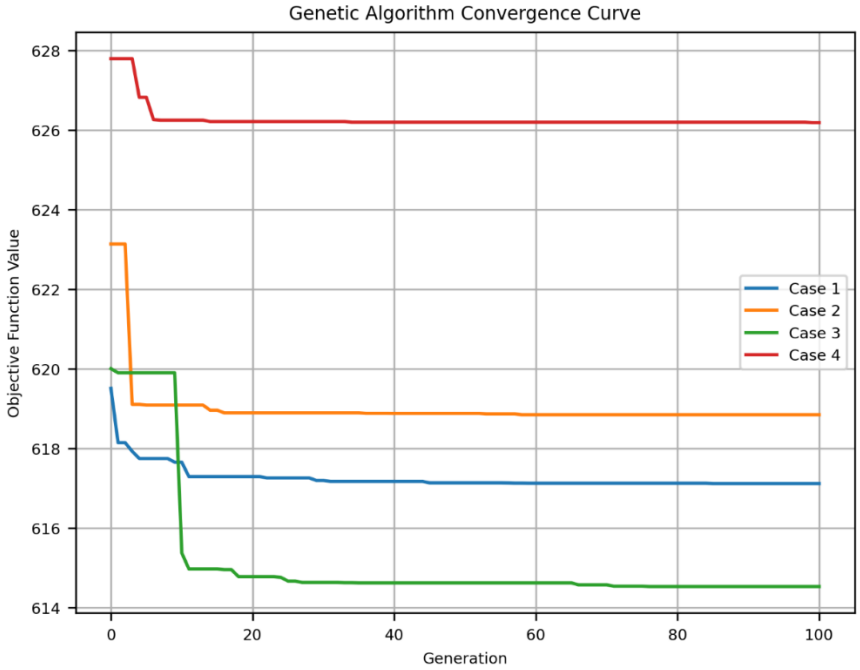

Fig. 9. The convergence curve of the GA application on four cases from the dataset with random forest regression as the fitness function

In our optimization, the features "Restaurant_latitude," "Restaurant_longitude," "Delivery_location_latitude," "Delivery_location_longitude," and "Type_of_order" are parameter constraints. While the "Delivery_person_Age," "Delivery_person_Ratings," "Type_of_vehicle," and "Time_taken(min)" features are the main objective parameters, and finally, "Delivery_person_ID" is the main objective. Table 2 compares the actual driver selection with the result of GA optimization. We chose these four cases because these four cases have the longest delivery time, which is 54 minutes. Parameter constraints make the restaurant location, customer location, and order the same for each case. The last column proves that GA optimization can decrease the delivery time for every four cases. The decreased delivery time was accompanied by a different selection of drivers for the four cases. In addition, GA optimization can also provide vehicle choices that streamline delivery time. The best efficiency is reducing the delivery time from 54 to 15 minutes.

Table 2. Comparison of The Actual and the GA Optimized Drive Selection

Case	Restaurant Lat, Long	Customer Lat, Long	Order Type	Actual Driver Selection (ID, Age, Rating, Vehicle, Time)	GA Optimized Driver Selection (ID, Age, Rating, Vehicle, Time)
1	26.9, 75.8	27.0, 75.9	Drinks	JAPRES12DEL02, 39, 4.1, motorcycle, 54	INDORES06DEL03, 18, 3, scooter, 15
2	22.7, 75.9	22.9, 6.0	Snack	INDORES12DEL02, 29, 4.6, motorcycle, 54	INDORES05DEL01, 50, 4, motorcycle, 27
3	26.9, 75.8	27.0, 75.9	Snack	JAPRES03DEL02, 32, 4.3, motorcycle, 54	INDORES07DEL02, 31, 2, motorcycle, 27
4	26.9, 75.8	27.0, 75.9	Meal	JAPRES05DEL02, 31, 5, motorcycle, 54	INDORES06DEL02, 49, 5, scooter, 28

3.2. Discussion

The strength of a correlation, as assessed by the Pearson correlation coefficient, is widely used to quantify the linear relationship between two variables. Several studies, such as research [45], stated that a value below 0.3 is generally considered indicative of a weak linear correlation. In the context of our online food delivery dataset, the Pearson correlation analysis revealed that the independent and dependent variables did not exhibit a linear relationship that surpassed the 0.3 threshold. While this might initially suggest a lack of association, it is important to note that this does not preclude the possibility of other types of relationships between the variables. As highlighted in another study [46], similar findings were observed, with no strong linear correlation apparent between the variables. Interestingly, despite this seemingly weak linear relationship, the study was still able to develop a regression model that demonstrated satisfactory performance. It underscores that linear correlation is not the sole determinant of predictive power.

The phenomenon observed in both the referenced study and our research suggests the presence of non-linear relationships between the variables. In these cases, the variables may be related in a more complex, non-linear manner that is not captured by the Pearson correlation coefficient. It highlights the importance of considering alternative relationships, such as polynomial or exponential, which may be better suited to describing the underlying interactions between the variables. In line with this understanding, our research yielded a random forest regression model with an impressive r^2 value of 0.98. This exceptional performance underscores the notion that strong predictive power can be achieved even without a strong linear correlation, further emphasizing the importance of exploring non-linear modeling techniques for accurately capturing complex relationships in the data.

The absence of multicollinearity among variables is crucial for the reliability and interpretability of regression models. Multicollinearity, the phenomenon of high correlation between independent variables, introduces several challenges in regression analysis [47]. Firstly, it leads to instability in coefficients, making it hard to accurately gauge the impact of each variable on the dependent variable due to their sensitivity to minor changes in the data. This instability can cause coefficients to fluctuate widely, diminishing their reliability in representing true relationships. Secondly, interpreting the contributions of correlated variables becomes challenging. Changes in one variable may be confounded by changes in another, obscuring which variable truly influences the observed effects.

Moreover, multicollinearity inflates standard errors of regression coefficients, heightening the risk of Type II errors, where important variables are erroneously labeled as non-significant [48]. Additionally, it leads to less reliable predictions, as the model needs help to discern the individual effects of highly correlated variables, potentially resulting in less accurate forecasts. Finally, identifying truly significant variables for predicting the dependent variable becomes difficult, impeding the processes of feature selection and model simplification. In our specific case, the absence of multicollinearity, as indicated by the results of the Pearson correlation test, is a positive sign for the reliability of our regression model. It suggests that the independent variables are not excessively correlated with each other, which means that the coefficients estimated by the model are likely to be more stable and interpretable. It, in turn, enhances the reliability of predictions and facilitates a clearer understanding of the individual contributions of each variable to the dependent variable.

Other main findings of our research are that applying GA and random forest regression as a fitness function significantly optimizes driver selection in the online food delivery system. Then, the proposed method demonstrates a substantial reduction in delivery time, with a notable efficiency improvement from 54 to 15 minutes. Subsequently, the random forest regression model outperforms other state-of-the-art regression models (linear regression, KNN, and AdaBoost) with an r^2 value of 0.98, RMSE of 54.3, and MAE of 11. These findings collectively highlight the transformative impact of our research on optimizing driver selection and delivery time in online food delivery systems.

Several studies have made predictions in the field of online food delivery using various regression methods such as linear regression [8], KNN regression [9], AdaBoost regression [10], and random forest [11]. On the other hand, other studies have also tried to optimize food delivery assignments using the multi-objective optimization method [49]. In contrast to [8]–[11], our research involves an optimization method, GA, in achieving optimum delivery time. On the other hand, the distinction between our research and [49] is using a regression model for the fitness function to optimize driver selection. In our research, we proved that random forest regression performs better than linear, KNN, and AdaBoost regression in predicting driver identity based on other variables in the online food delivery dataset. Our research contribution is a regression model in online food delivery that can predict driver identity using random forest regression.

Several studies have used GA and regression models as fitness functions for optimization in various fields, such as optimization in the field of cancer drugs [12], milling [13], copper nanocomposites [14], and casting system structures [15]. However, each of these studies has a research gap: they have yet to create a fitness

landscape that can better understand the complexity of optimization problems in fields that use GA as a solution. Drawing a fitness landscape of a GA based on the regression model's output serves as a visual representation of the problem's solution space. It provides valuable insights into how the model's performance varies across different combinations of input parameters. By examining the landscape, we can identify regions with high fitness (indicating optimal solutions) and areas with lower fitness (suboptimal solutions). This approach allows for a deeper understanding of how different combinations of input variables influence the model's performance.

Additionally, insight into the fitness landscape contour can guide the selection of appropriate optimization techniques and strategies. The second problem is that the application of GA in driver selection optimization in online food delivery is still a research opportunity. Our research contribution is two-fold. Our first contribution is the fitness landscape of random forest regression as a fitness function in a GA solution space using PCA dimension reduction, which can help understand the complexity of online food delivery optimization problems. Our second contribution is an optimization system for driver selection in online food delivery using GA and random forest.

By completing this research, we also achieve some practical implications. First, faster delivery times are a key factor in customer satisfaction in the online food delivery industry. Then, our algorithm can lead to happier customers who receive their orders more quickly [2]. Subsequently, optimizing driver selection can lead to more efficient use of resources, potentially reducing operational costs for the food delivery platform [50]. In addition, Implementing our algorithm could give the online food delivery platform a competitive edge in the market. At the same time, faster delivery times can be a strong selling point for attracting and retaining customers [51].

On the other hand, our research demonstrates the value of data-driven decision-making in the online food delivery industry. It can set a precedent for incorporating advanced algorithms and technology into business operations [52]. Lastly, the principles and techniques we have developed for optimizing driver selection may have broader applications in logistics and transportation beyond just food delivery [52]. Overall, our research has the potential to bring about positive changes in the online food delivery industry, benefiting both the platforms themselves and the customers they serve.

The study presents a novel approach for optimizing driver selection in the online food delivery system using Genetic Algorithms (GA) with random forest regression as the fitness function. However, it is important to acknowledge some limitations. The effectiveness of the proposed method may be contingent on the characteristics and scale of the specific dataset used. Future studies should explore its applicability across diverse datasets from various online food delivery platforms. Consequently, The random forest regression model may perform differently in different contexts. Assessing its generalization to various delivery scenarios, such as different city layouts or traffic conditions, is crucial. Then, the study may need to account for real-time factors that impact driver selection, such as sudden changes in order volume, traffic conditions, or unforeseen events. Future work could explore dynamic optimization strategies that adapt in real time. Lastly, while GA was used in this study, other optimization algorithms could offer alternative or complementary approaches. Comparative studies with different algorithms can provide valuable insights.

There is also some potential for future work. Future research could focus on integrating real-time data streams, such as traffic updates or order influx, to adjust driver selection for optimal results dynamically. Subsequently, extending the study to consider multiple conflicting objectives, like minimizing delivery time while maximizing driver utilization, would provide a more comprehensive optimization framework. Meanwhile, exploring driver behavior patterns and preferences could enhance the model by incorporating human-centric factors, potentially improving driver satisfaction and retention. Future work should also consider ethical considerations, like fairness in driver allocation, to ensure the optimization process aligns with societal values and norms. By addressing these limitations and pursuing these potential avenues for future research, the study can contribute even more substantially to online food delivery optimization.

4. CONCLUSION

Our research aim is to select the optimum driver for online food delivery using random forest regression and a genetic algorithm method. In achieving that aim, we have successfully implemented random forest regression and GA for driver selection in online food delivery. We use the Pearson correlation for feature analysis. In addition, we leverage linear regression, KNN regression, and AdaBoost Regression as benchmark methods for random forest regression. Finally, we carried out a fitness landscape analysis and GA parameter tuning and compared optimization results with actual results to prove GA performance. The test results show that random forest performs better than linear, KNN, and AdaBoost regression, with an r^2 value of 0.98, RMSE value of 54.3, and MAE value of 11. GA optimization can choose vehicles that streamline delivery time by

applying random forest regression as a fitness function in GA. The best efficiency is reducing the delivery time from 54 to 15 minutes. For future works, this research can be directed to several topics, including GA simulation, deployment and testing, ethical and privacy examinations, user experience, and system integration into the online delivery system. By completing this research, we also achieve some practical implications, such as faster delivery times, a key factor in customer satisfaction in the online food delivery industry. The first key is an optimum driver selection model in random forest regression, while the second is an optimum driver selection model in GA. Future work should also consider ethical considerations, like fairness in driver allocation, to ensure the optimization process aligns with societal values and norms. Conducting real-world experiments or simulations to validate the proposed optimization method in practical settings would bolster the study's applicability and reliability. This study introduces a groundbreaking approach to driver selection optimization in online food delivery, combining GA and random forest regression, with the potential to revolutionize the industry, enhance customer satisfaction, and pave the way for future advancements in the field. In contemplating the future of online food delivery, our study underscores the immense potential for data-driven strategies to redefine industry standards and enhance the customer experience, inviting us to envision a landscape where every delivery is not just efficient but optimized to perfection, setting a new benchmark for excellence in the realm of service delivery.

Acknowledgments

We thank the Directorate of Research and Community Service (PPM) Telkom University for fully funding this research. We also thank Bhanupratap Biswas for his research on online food delivery in India and for sharing their dataset on Kaggle.

REFERENCES

[1] W. Yao, H. Zhao, and L. Liu, "Weather and time factors impact on online food delivery sales: a comparative analysis of three Chinese cities," *Theor. Appl. Climatol.*, pp. 1–14, 2023, https://doi.org/10.1007/s00704-023-04542-w.

[2] F. Huq, N. Sultana, and M. A. Razzaque, "Quality of Service Aware Order Allocation for Inter-Regional Online Food Delivery Systems," in *25th International Conference on Advanced Communication Technology (ICACT)*, 2023, pp. 358–364, 2023, https://doi.org/10.23919/ICACT56868.2023.10079492.

[3] D. Gavilan, A. Balderas-Cejudo, S. Fernández-Lores, and G. Martinez-Navarro, "Innovation in online food delivery: Learnings from COVID-19," *Int. J. Gastron. Food Sci.*, vol. 24, p. 100330, 2021, https://doi.org/10.1016/j.ijgfs.2021.100330.

[4] A. T. Saad, "Factors affecting online food delivery service in Bangladesh: an empirical study," *Br. Food J.*, vol. 123, no. 2, pp. 535–550, 2021, https://doi.org/10.1108/BFJ-05-2020-0449.

[5] Y. T. Prasetyo *et al.*, "Factors affecting customer satisfaction and loyalty in online food delivery service during the COVID-19 pandemic: Its relation with open innovation," *J. Open Innov. Technol. Mark. Complex.*, vol. 7, no. 1, p. 76, 2021, https://doi.org/10.3390/joitmc7010076.

[6] M. Franke and V. Pulignano, "Connecting at the edge: Cycles of commodification and labour control within food delivery platform work in Belgium," *New Technol. Work Employ.*, vol. 38, no. 2, pp. 371–390, 2023, https://doi.org/10.1111/ntwe.12218.

[7] S. F. Pane, A. G. Putrada, N. Alamsyah, and M. N. Fauzan, "A PSO-GBR Solution for Association Rule Optimization on Supermarket Sales," in *Seventh International Conference on Informatics and Computing (ICIC)*, pp. 1–6, 2022, https://doi.org/10.1109/ICIC56845.2022.10007001.

[8] M. Torabbeigi, G. J. Lim, and S. J. Kim, "Drone delivery scheduling optimization considering payload-induced battery consumption rates," *J. Intell. Robot. Syst.*, vol. 97, pp. 471–487, 2020, https://doi.org/10.1007/s10846-019-01034-w.

[9] S. Hughes, S. Moreno, W. F. Yushimito, and G. Huerta-Cánepa, "Evaluation of machine learning methodologies to predict stop delivery times from GPS data," *Transp. Res. Part C Emerg. Technol.*, vol. 109, pp. 289–304, 2019, https://doi.org/10.1016/j.trc.2019.10.018.

[10] K. Wang and L. Liu, "Design of Natural Human-Computer Interaction for Unmanned Delivery Vehicle Based on Kinect," in *HCI in Mobility, Transport, and Automotive Systems*, pp. 156–169, 2021, https://doi.org/10.1007/978-3-030-78358-7_10.

[11] H. Errousso, N. Malhene, S. Benhadou, and H. Medromi, "Predicting car park availability for a better delivery bay management," *Procedia Comput. Sci.*, vol. 170, pp. 203–210, 2020, https://doi.org/10.1016/j.procs.2020.03.026.

[12] M. Salehi, S. Farhadi, A. Moieni, N. Safaie, and M. Hesami, "A hybrid model based on general regression neural network and fruit fly optimization algorithm for forecasting and optimizing paclitaxel biosynthesis in Corylus avellana cell culture," *Plant Methods*, vol. 17, pp. 1–13, 2021, https://doi.org/10.1186/s13007-021-00714-9.

[13] A. Yeganefar, S. A. Niknam, and R. Asadi, "The use of support vector machine, neural network, and regression analysis to predict and optimize surface roughness and cutting forces in milling," *Int. J. Adv. Manuf. Technol.*, vol. 105, pp. 951–965, 2019, https://doi.org/10.1007/s00170-019-04227-7.

[14] A. Torabi, R. M. Babaheydari, G. H. Akbari, and S. O. Mirabootalebi, "Optimizing of micro-hardness of nanostructured Cu–Cr solid solution produced by mechanical alloying using ANN and genetic algorithm," *SN Appl. Sci.*, vol. 2, pp. 1–9, 2020, https://doi.org/10.1007/s42452-020-03722-x.
[15] H. Chen, Q. Gao, Z. Wang, Y. Fan, W. Li, and H. Wang, "Optimization of Casting System Structure Based on Genetic Algorithm for A356 Casting Quality Prediction," *Int. J. Met.*, vol. 17, no. 3, pp. 1948–1969, 2023, https://doi.org/10.1007/s40962-022-00902-w.
[16] C. Rojon, A. McDowall, and M. N. K. Saunders, "The Relationships Between Traditional Selection Assessments and Workplace Performance Criteria Specificity: A Comparative Meta-Analysis," *Hum. Perform.*, vol. 28, no. 1, pp. 1–25, 2015, https://doi.org/10.1080/08959285.2014.974757.
[17] F. H. Hidayatullah, M. Abdurohman, and A. G. Putrada, "Accident Detection System for Bicycle Athletes Using GPS/IMU Integration and Kalman Filtered AHRS Method," in *International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS)*, pp. 1–6, 2021, https://doi.org/10.1109/ICADEIS52521.2021.9702085.
[18] P. R. Kannari, N. C. Shariff, and R. L. Biradar, "Network intrusion detection using sparse autoencoder with swish-PReLU activation model," *J. Ambient Intell. Humaniz. Comput.*, pp. 1–13, 2021, https://doi.org/10.1007/s12652-021-03077-0.
[19] G. Zeng, "On the analytical properties of category encodings in logistic regression," *Commun. Stat. - Theory Methods*, vol. 52, no. 6, pp. 1870–1887, 2023, https://doi.org/10.1080/03610926.2021.1939382.
[20] A. K. Tripathi, G. Rathee, and H. Saini, "Taxonomy of missing data along with their handling methods," in *Fifth International Conference on Image Information Processing (ICIIP)*, pp. 463–468, 2019, https://doi.org/10.1109/ICIIP47207.2019.8985715.
[21] M. B. Satrio, A. G. Putrada, and M. Abdurohman, "Evaluation of Face Detection and Recognition Methods in Smart Mirror Implementation," in *Proceedings of Sixth International Congress on Information and Communication Technology*, pp. 449–457, 2022, https://doi.org/10.1007/978-981-16-2380-6_39.
[22] A. G. Putrada and D. Perdana, "Improving thermal camera performance in fever detection during covid-19 protocol with random forest classification," in *International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS)*, pp. 1–6, 2021, https://doi.org/10.1109/ICADEIS52521.2021.9702045.
[23] Z. Zhu, P. Zhang, Z. Liu, and J. Wang, "Static Voltage Stability Assessment Using a Random UnderSampling Bagging BP Method," *Processes*, vol. 10, no. 10, p. 1938, 2022, https://doi.org/10.3390/pr10101938.
[24] F. Harrou, A. Saidi, and Y. Sun, "Wind power prediction using bootstrap aggregating trees approach to enabling sustainable wind power integration in a smart grid," *Energy Convers. Manag.*, vol. 201, p. 112077, 2019, https://doi.org/10.1016/j.enconman.2019.112077.
[25] M. D. Nastiti, M. Abdurohman, and A. G. Putrada, "Smart shopping prediction on smart shopping with linear regression method," in *7th International Conference on Information and Communication Technology (ICoICT)*, 2019, pp. 1–6, 2019, https://doi.org/10.1109/ICoICT.2019.8835271.
[26] M. Korkmaz, "A study over the general formula of regression sum of squares in multiple linear regression," *Numer. Methods Partial Differ. Equ.*, vol. 37, no. 1, pp. 406–421, 2021, https://doi.org/10.1002/num.22533.
[27] A. G. Putrada, M. Abdurohman, D. Perdana, and H. H. Nuha, "EdgeSL: Edge-Computing Architecture on Smart Lighting Control With Distilled KNN for Optimum Processing Time," *IEEE Access*, vol. 11, pp. 64697–64712, 2023, https://doi.org/10.1109/ACCESS.2023.3288425.
[28] Y. Zhou, M. Huang, and M. Pecht, "Remaining useful life estimation of lithium-ion cells based on k-nearest neighbor regression with differential evolution optimization," *J. Clean. Prod.*, vol. 249, p. 119409, 2020, https://doi.org/10.1016/j.jclepro.2019.119409.
[29] A. G. Putrada and D. Perdana, "Improving Thermal Camera Performance in Fever Detection during COVID-19 Protocol with Random Forest Classification," *International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS)*, pp. 1-6, 2021, https://doi.org/10.1109/ICADEIS52521.2021.9702045.
[30] M. N. Fauzan, A. G. Putrada, N. Alamsyah, and S. F. Pane, "PCA-AdaBoost Method for a Low Bias and Low Dimension Toxic Comment Classification.," in *International Conference on Advanced Creative Networks and Intelligent Systems (ICACNIS)*, pp. 1–6, 2022, https://doi.org/10.1109/ICACNIS57039.2022.10055017.
[31] W. Jianlong, S. H. Jaaman, and H. B. Samsudin, "R-squared measurement in multifactor pricing model," in *AIP Conference Proceedings*, vol. 1678, no. 1, 2015, https://doi.org/10.1063/1.4931328.
[32] M. Erdt, A. Fernández and C. Rensing, "Evaluating Recommender Systems for Technology Enhanced Learning: A Quantitative Survey," in *IEEE Transactions on Learning Technologies*, vol. 8, no. 4, pp. 326-344, 2015, https://doi.org/10.1109/TLT.2015.2438867.
[33] K. A. Beyene, "Comparative study of linear and quadratic model equations for prediction and evaluation of surface roughness of a plain-woven fabric," *Res. J. Text. Appar.*, vol. 27, no. 2, pp. 281–298, 2023, https://doi.org/10.1108/RJTA-08-2021-0107.
[34] A. Subasi, M. F. El-Amin, T. Darwich, and M. Dossary, "Permeability prediction of petroleum reservoirs using stochastic gradient boosting regression," *J. Ambient Intell. Humaniz. Comput.*, pp. 1–10, 2020, https://doi.org/10.1007/s12652-020-01986-0.
[35] R. R. Maaliw, M. A. Ballera, Z. P. Mabunga, A. T. Mahusay, D. A. Dejelo, and M. P. Seño, "An ensemble machine learning approach for time series forecasting of COVID-19 cases," in *IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0633–0640, 2021, https://doi.org/10.1109/IEMCON53756.2021.9623074.

[36] J. Wieczorek, C. Guerin, and T. McMahon, "K-fold cross-validation for complex sample surveys," *Stat*, vol. 11, no. 1, p. e454, 2022, https://doi.org/10.1002/sta4.454.
[37] A. G. Putrada, M. Abdurohman, D. Perdana, and H. H. Nuha, "CIMA: A Novel Classification-Integrated Moving Average Model for Smart Lighting Intelligent Control Based on Human Presence," *Complexity*, vol. 2022, pp. 1–19, 2022, https://doi.org/10.1155/2022/4989344.
[38] B. Alhijawi and A. Awajan, "Genetic algorithms: Theory, genetic operators, solutions, and applications," *Evol. Intell.*, pp. 1–12, 2023, https://doi.org/10.1007/s12065-023-00822-6.
[39] R. Karmakar, "Application of Genetic Algorithm (GA) in Medical Science: A Review," in *Second International Conference on Sustainable Technologies for Computational Intelligence: Proceedings of ICTSCI 2021*, pp. 83–94, 2021, https://doi.org/10.1007/978-981-16-4641-6_8.
[40] M. Zare, F. Pazooki, and S. E. Haghighi, "Hybrid controller of Lyapunov-based and nonlinear fuzzy-sliding mode for a quadrotor slung load system," *Eng. Sci. Technol. Int. J.*, vol. 29, p. 101038, 2022, https://doi.org/10.1016/j.jestch.2021.07.001.
[41] D. M. Chitty, "An ant colony optimisation inspired crossover operator for permutation type problems," in *2021 IEEE Congress on Evolutionary Computation (CEC)*, pp. 57–64, 2021, https://doi.org/10.1109/CEC45853.2021.9504893.
[42] R. D. Goswami, S. Chakraborty, and B. Misra, "Variants of Genetic Algorithms and Their Applications," in *Applied Genetic Algorithm and Its Variants: Case Studies and New Developments*, pp. 1–20, 2023, https://doi.org/10.1007/978-981-99-3428-7_1.
[43] D. Diaz Martinez, R. Trujillo Codorniu, R. Giral, and L. Vazquez Seisdedos, "Evaluation of particle swarm optimization techniques applied to maximum power point tracking in photovoltaic systems," *Int. J. Circuit Theory Appl.*, vol. 49, no. 7, pp. 1849–1867, 2021, https://doi.org/10.1002/cta.2978.
[44] T. Avdeenko and K. Serdyukov, "Genetic Algorithm Fitness Function Formulation for Test Data Generation with Maximum Statement Coverage," in *Advances in Swarm Intelligence*, vol. 12689, pp. 379–389, 2021, doi: 10.1007/978-3-030-78743-1_34.
[45] X. Gao, X. Li, B. Zhao, W. Ji, X. Jing, and Y. He, "Short-term electricity load forecasting model based on EMD-GRU with feature selection," *Energies*, vol. 12, no. 6, p. 1140, 2019, https://doi.org/10.3390/en12061140.
[46] C. Kaup, "The optimum of heat recovery-Determination of the optimal heat recovery based on a multiple non-linear regression model," *J. Build. Eng.*, vol. 38, p. 101548, 2021, https://doi.org/10.1016/j.jobe.2020.101548.
[47] X. Feng, D. S. Park, Y. Liang, R. Pandey, and M. Papeş, "Collinearity in ecological niche modeling: Confusions and challenges," *Ecol. Evol.*, vol. 9, no. 18, pp. 10365–10376, 2019, https://doi.org/10.1002/ece3.5555.
[48] B. G. Domb and P. W. Sabetian, "The blight of the type II error: when no difference does not mean no difference," *Arthrosc. J. Arthrosc. Relat. Surg.*, vol. 37, no. 4, pp. 1353–1356, 2021, https://doi.org/10.1016/j.arthro.2021.01.057.
[49] Z. Lou, W. Jie, and S. Zhang, "Multi-objective optimization for order assignment in food delivery industry with human factor considerations," *Sustainability*, vol. 12, no. 19, p. 7955, 2020, https://doi.org/10.3390/su12197955.
[50] Y. A. K. Reddy, C. S. Swaroop, S. Terence, K. S. M. Reddy, and K. Santhosh, "Zero Cost Online Food Delivery System with Machine Learning Prediction," in *7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 407–412, 2023, https://doi.org/10.1109/ICICCS56967.2023.10142721.
[51] M. Keeble, J. Adams, and T. Burgoine, "Investigating experiences of frequent online food delivery service use: a qualitative study in UK adults," *BMC Public Health*, vol. 22, no. 1, p. 1365, 2022, https://doi.org/10.1186/s12889-022-13721-9.
[52] G. Zou and Y. Li, "Improvement of genetic algorithm and the application in computer simulation model of O2O delivery strategies," in *13th International Symposium on Computational Intelligence and Design (ISCID)*, pp. 295–299, 2020, https://doi.org/10.1109/ISCID51228.2020.00072.

BIOGRAPHY OF AUTHORS

Aji Gautama Putrada is enrolled in Telkom University's Doctor of Philosophy in Computer Science program. His dissertation explores user comfort in machine learning-based smart lighting. Email: ajigps@telkomuniversity.ac.id.

Nur Alamsyah, is enrolled in Telkom University's Doctor of Philosophy in Computer Science program. His dissertation explores Airfare Dynamic Pricing Based On Sentiment Analysis. Email: nuralamsyah@student.telkomuniversity.ac.id.

Ikke Dian Oktaviani, is enrolled in Telkom University's Doctor of Philosophy in Computer Science program. Her research is about IoT stream data processing.
Email: oktavianiid@telkomuniversity.ac.id.

Mohamad Nurkamal Fauzan is enrolled in the Doctor of Philosophy at Telkom University in the Computer Science program. He is interested in IoT and machine learning. Email: mnurkamalfauzan@student.telkomuniversity.ac.id.