Optimizing Random Forest Models with Snake Optimization Algorithm for Predicting E-commerce User Purchase Behaviour

Pengfei Li
School of Management, Zhengzhou Business University, Gongyi City, Henan Province, China
E-mail: Lee_samuelson@163.com

Keywords: e-commerce users, purchase behaviour, random forest, snake optimization algorithm

Recieved: December 5, 2024

This study proposes a Snake Optimization-based Random Forest (SO-RF) model for predicting e-commerce user behavior. Key user interaction metrics, including browsing records, purchase history, search keywords, click rate, dwell time, add-to-cart times, user comments, and visit time, serve as input features, while user conversion rate and purchase rate are the target metrics. The dataset undergoes preprocessing and feature engineering to extract meaningful patterns. The Snake Optimization (SO) algorithm fine-tunes the hyperparameters of the Random Forest (RF) model, enhancing predictive performance and generalization. Experimental results demonstrate that SO-RF outperforms conventional RF, Simulated Annealing-based RF (SA-RF), and Sparrow Search Algorithm-based RF (SSA-RF) on the test set, achieving an MAE of 0.31959, MAPE of 1.6652, MSE of 0.17625, RMSE of 0.41983, and R^2 of 0.96678. These findings provide valuable insights for e-commerce platforms, enabling personalized marketing strategies, improved user experience, and enhanced sales performance through accurate behavior prediction.

Povzetek: Članek predstavi optimiziran model naključnih gozdov z algoritmom Snake (SO-RF) za napovedovanje nakupnega vedenja uporabnikov e-trgovine, s čimer izboljša prodajo in personalizacijo.

1 Introduction

With the rapid development of e-commerce, e-commerce platforms have accumulated a large amount of user behavior data (Xie C, 2020). These data contain information about users' browsing records, purchase history, search keywords, click rate, dwell time, number of shopping carts added, user comments, visit time, and other aspects of e-commerce platforms. Through the analysis and mining of these data, we can gain an in-depth understanding of users' shopping habits, needs, and preferences, which provides the basis for enterprises to develop more accurate and personalized marketing strategies (Wu Z, 2021). Therefore, e-commerce user behavior prediction research is of great significance.

First of all, e-commerce user behavior prediction research can help enterprises better understand user needs and market conditions (Khrais L T, 2020). Through the analysis and mining of user behavior data, we can understand the shopping habits, needs, and preferences of users and grasp the changes and trends of the market. This information can help enterprises develop more accurate and personalized marketing strategies, improve user conversion and purchase rates, and increase their sales and profits (Niu Z, 2021). At the same time, through the analysis of user behavior data, the potential needs and unsatisfied needs of users can also be found, providing a basis for enterprises to develop new products and services (Blazevic V, 2008).

Secondly, e-commerce user behavior prediction research can improve the marketing efficiency and effectiveness of enterprises (Wakil K, 2020). Through the analysis and mining of user behavior data, target user groups can be accurately located, and targeted marketing strategies can be formulated. Relevant goods and services can be recommended to users based on their browsing records and purchase history; the titles and descriptions of goods can be optimized based on users' search keywords and click-through rates to increase the click-through and purchase rates of goods; and the design and layout of the page can be optimized based on the user's dwell time and the number of times they add to the shopping cart to improve the user's shopping experience and satisfaction. These targeted marketing strategies can improve the marketing efficiency and effectiveness of enterprises, reduce marketing costs, and increase their sales and profits (Zhang B, 2021).

Again, e-commerce user behavior prediction research can improve the competitiveness and market share of enterprises (Xiahou X, 2022). Through the analysis and mining of user behavior data, changes and trends in the market can be found, providing the basis for enterprises to develop more accurate and personalized marketing strategies. These strategies can help enterprises stand out in the fierce market competition and improve their competitiveness and market share (To M L, 2006). At the same time, through the analysis of user behavior data, new market opportunities, and business models can also be found, providing new ideas and directions for the

development of enterprises (Chang K, 2003).

Finally, e-commerce user behavior prediction research can also provide valuable data support and a decision-making basis for enterprises (Fan S, 2015). Through the analysis and mining of user behavior data, it can provide valuable data support and a decision-making basis for enterprises. These data can provide support and guidance for strategic planning, product development, marketing, and other aspects of the enterprise, helping the enterprise to make more scientific and reasonable decisions (Poggi N, 2013). At the same time, through the analysis of user behavior data, the problems and shortcomings of enterprises can also be found, providing a basis for enterprises to improve and perfect their services (Guo Y, 2018).

E-commerce user behavior data contains a large amount of information, how to extract meaningful features from this data to better understand the user needs and market conditions is an important problem to be solved in this study. The random forest model is a commonly used machine learning algorithm, but its performance is affected by parameter settings. How to optimize the parameters of the random forest model using a snake optimization algorithm to improve the performance and generalization ability of the model. The accuracy and reliability of the prediction results are important metrics to assess the performance of the model. The goal of this study is to predict user behavior—specifically conversion rate and purchase rate—in order to assist businesses better comprehend user requirements and market conditions, create more precise and personalized marketing tactics, and increase user satisfaction and sales. The chosen metrics—dwell time, click rate, purchase history, and search keywords—are crucial for comprehending user shopping behavior and improving predictive accuracy. Dwell time reflects user engagement, as more time spent on a product page indicates increased interest and purchase intent. Click rate shows interaction frequency, revealing which products receive the most attention. Purchase history offers direct insights into past purchasing patterns, allowing you to forecast future purchases using established preferences. Search keywords reveal user intent by emphasizing particular interests and requirements at various stages of the purchasing process. These metrics were selected using previous research and industry best practices, ensuring their usefulness in accurately forecasting conversion and purchase rates. Their inclusion improves model efficiency by capturing both explicit and implicit customer behavior signals, rendering them the ideal option for improving personalized suggestions and marketing tactics.

The Snake Optimization (SO) algorithm is ideal for improving Random Forest (RF) in e-commerce analytics because of its adaptive exploration-exploitation balance, which effectively tunes hyperparameters such as tree depth and the number of estimators to improve predictive accuracy. Unlike traditional optimization techniques, SO simulates natural selection behaviors, resulting in a more varied and broadly optimal parameter search, which is critical for dealing with complex, high-dimensional e-commerce data. Beyond e-commerce, SO-RF can be used in healthcare analytics to optimize diagnostic models by fine-tuning disease prediction classification thresholds, as well as marketing analytics to improve consumer segmentation models by improving decision trees using consumer behavior data. The algorithm's versatility in feature selection and parameter optimization renders it an effective tool for a wide range of predictive modeling tasks across industries.

1.1 Research objective

This research aims to improve e-commerce user behavior prediction by combining SO with Random Forest (RF), resulting in the SO-RF model. The main objective is to enhance prediction accuracy, optimize feature selection, and decrease model bias-variance trade-offs using adaptive parameter tuning.

1.2 Hypotheses

H1: SO-RF attains higher predictive accuracy than conventional RF, Simulated Annealing (SA-RF), and Sparrow Search Algorithm (SSA-RF).

H2: SO's adaptive search mechanism enhances feature selection, leading to better generalization and decreased overfitting.

H3: Despite its computational complexity, SO improves model robustness in managing dynamic user behavior patterns in e-commerce.

1.3 Expected outcomes for E-commerce applications

The proposed SO-RF model is expected to offer more accurate predictions of user behavior, allowing e-commerce platforms to improve customized suggestions, enhance marketing tactics, and increase customer engagement. Furthermore, the model's scalability and adaptability may enable dynamic pricing, demand prediction, and real-time decision-making in online retail settings.

2 Related study

Random Forest (RF) is an integrated learning model that improves the overall prediction accuracy by combining the prediction results of multiple decision trees (Naghibi S A, 2016). Random forest models have a wide range of research areas, including classification: random forests can be applied to multi-class classification problems, such as image classification, text classification, bioinformatics classification, etc. (Verikas A, 2011);

regression: random forests can be applied to regression problems, such as house price prediction, stock price prediction, etc. (Fawagreh K, 2014); feature selection: random forests can provide importance ranking of features and thus help in feature selection (Cutler D R, 2007). Anomaly detection: random forests can be applied to anomaly detection, such as financial fraud detection, network intrusion detection, etc. (Amaratunga D, 2008). Bioinformatics: Random Forest can be applied to bioinformatics problems such as gene classification, protein structure prediction, etc (Segal M, 2011). Marketing: random forests can help companies analyze customer data to develop more accurate marketing strategies (Li T, 2016). Healthcare: random forests can be applied to healthcare problems such as disease risk prediction and patient susceptibility prediction (Bin J, 2016).

The main advantages of the model include: efficient training speed and good parallelization ability; strong robustness to high-dimensional data and missing data; the ability to give the importance ranking of features, which helps feature selection; and good generalization ability and resistance to overfitting (Ao Y, 2019). However, the random forest model also has some defects: it is more sensitive to noise and outliers; in some cases, overfitting may occur; for some specific types of data, such as text data or image data, special preprocessing or feature engineering may be required; and the interpretability of the model is relatively poor, which makes it difficult to intuitively understand the decision-making process of the model (Shi K, 2018).

To address the shortcomings of the random forest model, there are some improvement methods, such as the introduction of regularization terms and the use of deep forest structure (Zhang W, 2021). In addition, some studies are focusing on how to improve the interpretability of the random forest model, such as decision tree-based interpretation methods, model decomposition-based interpretation methods, etc. (Ren S, 2015).

Yan and Zhou (2024) proposed a recommendation algorithm that uses matrix reduction methods to improve network information analysis and user behavior predictions. Yuan (2024) proposed a deep learning-based framework for predicting consumer behavior, which uses sophisticated neural networks to optimize enterprise precision marketing campaigns. Cheng and He (2024) used random forest optimization to improve product modeling processes, improving design effectiveness and visual efficiency in e-commerce applications. Table 1 shows the summary table of these existing works.

Table 1: Summary table

Citation	Key Focus	Key Findings	Applications
Naghibi et al. (2016)	Groundwater potential mapping utilizing RF, BRT, and CART	BRT surpassed CART and RF with an AUC of 0.8103, while RF had the minimum at 0.7119	GIS-based environmental tracking
Verikas et al. (2011)	Variable significance and performance of RF	Found high variance in variable importance rankings, denoting instability in small datasets	Feature selection and data exploration
Fawagreh et al. (2014)	Evolution and improvements in RF	Discussed RF's enhancements and future directions in ensemble learning	Classification and data mining
Cutler et al. (2007)	RF for ecological classification	RF demonstrated better classification accuracy compared to conventional techniques.	Ecological modeling and species classification
Amaratunga et al. (2008)	Enriched RF for feature selection	Proposed weighted sampling to enhance RF performance in high-dimensional datasets	Bioinformatics and microarray data evaluation
Segal & Xiao (2011)	Multivariate RF for numerous response prediction	Showed improved predictive accuracy in multi-response settings	Ecology and predictive modeling
Bin et al. (2016)	Modified RF for multi-class classification	Enhanced RF's performance utilizing NIR spectroscopy for tobacco leaf grading	Spectroscopy-based classification

Existing RF-based methodologies and variants have limitations like instability in feature selection, bias toward dominant features, and inadequacies in dealing with high-dimensional or imbalanced data. While improvements such as enriched RF and multivariate RF tackle some problems, they do not include adaptive learning or optimal feature organization. A hybrid SO-RF approach addresses these gaps by incorporating self-organizing mechanisms, which improve feature selection, adaptability, and classification efficiency, rendering it more resilient for intricate datasets.

3 Description Of the methodology

3.1 Random Forest model

The random forest model is studied because it performs well in many real-world problems and has the advantages of being efficient, robust, and easy to use. Random forest model is an integrated learning model based on decision trees, which improves the overall prediction accuracy by combining the prediction results of multiple decision trees. Its core idea is to use the self-service sampling (bootstrap sampling) method to extract multiple samples from the original dataset, and then construct a decision tree for each sample, and ultimately vote or average the prediction results of all the decision trees to arrive at the final prediction results (Lin W, 2017). It can handle classification and regression problems effectively by constructing multiple decision trees and integrating them. The interpretability of Random Forest is known to be limited because it functions as an ensemble of decision trees, rendering it hard to directly explain individual predictions. To tackle this, techniques such as SHAP (SHapley Additive Explanations) values can be used to quantify each feature's contribution to the model's decisions. By incorporating SHAP analysis, the model's results can be better understood, providing insights into feature importance and decision-making procedures, ultimately enhancing transparency and trust in predictions.

3.1.1 Decision tree formulation

Decision tree is the basic component of a random forest, and its formula includes the following aspects:

Finding the information gain formula:

The information gain is used to measure the degree of information reduction under the division of feature values, and its formula is:

$$\Delta H(D,A) = H(D) - \sum_{v=1}^{V} \frac{|D_v|}{|D|} H(D_v) \quad (1)$$

where is the initial information entropy of the dataset, is the conditional entropy when the feature takes the value of, is the number of values of the feature, is the number of samples of the dataset, is the number of samples when the feature takes the value of.

Find the formula for the Gini index:

The Gini index is used to measure the purity of the dataset with the formula:

$$Gini(D) = 1 - \sum_{k=1}^{K} (P_k)^2 \quad (2)$$

Where K is the number of categories in the dataset and P_k is the proportion of samples in the dataset belonging to category No. to the total samples.

Decision tree construction algorithm formula

The decision tree construction algorithm is usually based on information gain or the Gini index for feature selection. The formula for building a decision tree is as follows:

Input: training set, feature set, threshold value

Output: decision tree

If the samples all belong to the same category, it will be returned as a single node tree, labeled as;

If it is the empty set, i.e., there are no more features to choose from, it will be treated as a single node tree, labeled as the category with the highest number of samples in it, and returned;

Select the optimal feature based on the information gain or Gini index;

If the information gain or Gini index is less than the threshold, it will be returned as a single node tree, labeled as the category with the highest number of samples in;

Otherwise, it will be divided into subsets based on the values of the features;

For each subset, recursively call the above steps to construct the subtree;

Will be connected to the top.

3.1.2 Random Forest formulation

Random forest is an algorithm for prediction or classification by integrating multiple decision trees, and its formula includes the following aspects:

Random forest generation formula:

The formula for random forest generation is:

$$RF(X) = \frac{1}{T} \sum_{t=1}^{T} f_t(X) \quad (3)$$

where $RF(X)$ denotes the prediction result of the random forest on the sample X, T denotes the number of decision trees in the random forest, and $f_t(X)$ denotes the prediction result of the tth decision tree on the sample X.

Feature selection formula:

Random forests are constructed by randomly selecting features for decision tree construction, and the formula for feature selection is:

$$S = \sum_{i=1}^{S} i\,\text{importance(i)} \quad (4)$$

where S denotes the sum of selection probabilities of all features in the feature set and $importance(i)$ Denotes the selection probability of the feature.

3.2 Snake optimization algorithm

Snake Optimization Algorithm (SO) is an intelligent optimization algorithm that simulates the specific mating behavior of snakes, proposed by Fatma A. Hashim and Abdelazim G. Hussien in 2022, which is inspired by the foraging and reproductive behaviors and patterns of snakes (Hashim F A. 2022).

The principle of the snake optimization algorithm is to simulate the mating behavior of snakes in late spring and early summer. The mating process of snakes depends not only on the temperature but also on the availability of food. If the temperature is low and food is sufficient, mating occurs, otherwise the snake will only search for food or eat the remaining food. In the snake optimization algorithm, the population is divided into two equal groups i.e. males and females. Male snakes will fight with each other to attract the attention of females. Females have the right to decide whether to mate or not. If mating occurs, the female starts laying eggs in the nest or burrow and once the eggs appear, she leaves (Zheng W, 2023).

The snake optimization algorithm is divided into two stages, namely global exploration and local exploitation. Exploration represents the environmental factors, i.e., cold places and food, in which case the snake only searches for food in its surroundings. For exploitation, this phase includes many transitions to make the global more efficient. In situations where food is available but the temperature is high, the snake will only focus on eating the available food. In the battle mode, each male will fight to get the best female and each female will try to choose the best male (Fu H, 2022). Snake optimization algorithm is a kind of intelligent optimization algorithm that simulates the behavior of natural creatures, with good optimization-seeking ability and fast convergence. It can be applied to a variety of practical problems, such as function optimization, machine learning, deep learning, and so on (Yan C, 2023).

The snake optimization algorithm first generates a uniformly distributed random totality to be able to start the optimization algorithm process.

$$X_i = X_{\min} + r \times (X_{\max} - X_{\min}) \tag{5}$$

It will be divided into two groups of males and females, and for the study, it is assumed that the number of males will be 50% and the number of females will be 50%. The exploration and development phase of the snake optimization algorithm is mainly affected by the temperature Temp and the amount of food Q, which is given by Eq (6)

$$Temp = \exp(-\frac{t}{T}), \quad Q = c_1 \times \exp(\frac{t-T}{T}) \tag{6}$$

where t represents the current iteration, T represents the maximum number of iterations, and c1=0.5. Exploration phase (no food): Q<threshold (0.25); exploitation phase (food present): Q>threshold (0.6).

When Q<threshold (0.25), the stochastic exploration formula is:

$$X_i^m = X_{rand}^m(t) \pm c_2 \times A_m \times ((X_{\max} - X_{\min}) \times rand + X_{\min}) \tag{7}$$

where X_i^m to male snake location, X_{rand}^m refers to random snake location, c1 = 0.05, and A_m is the ability of males to find.

The Snake Optimization Algorithm (SO) converts snake mating behavior into an effective optimization framework by modeling the balance between exploration (searching for optimal solutions) and exploitation (fine-tuning promising candidates). In machine learning, SO efficiently tunes model parameters by utilizing its dual-phase method: the exploration phase, driven by temperature and food availability, guarantees a diverse search of the solution space to avoid premature convergence, while the exploitation phase, where male snakes compete for females, improves the best solutions through selective mating. This adaptive mechanism, which simulates evolutionary choice, enables SO to dynamically adjust learning rates, feature weights, and hyperparameters in machine learning models. SO improves convergence speed and accuracy by integrating biologically inspired search and selection procedures, rendering it an effective tool for parameter tuning in intricate optimization tasks.

3.3 Snake optimization algorithm to optimize random forest model

The snake optimization algorithm can be applied to optimize the parameters of the random forest model. The following are the steps, formulas, and principles for optimizing the random forest model:

Steps:

Initialize the parameters of the snake optimization algorithm, including the number of populations, the maximum number of iterations, the dimensionality, and so on.

Use the snake optimization algorithm to generate uniformly distributed random populations as the initial parameters of the random forest model.

Calculate the fitness value of each individual according to the performance evaluation indexes (such as accuracy, recall, etc.) of the random forest model.

According to the principle of the snake optimization algorithm, the population is updated and evolved, including two stages of global exploration and local development.

Repeat steps 3 and 4 until the maximum number of iterations is reached or the stopping condition is satisfied.

Output the optimal random forest model parameters.

Formula:

In the snake optimization algorithm, the position update formula for each individual is:

$$X_{i,j(t+1)} = X\_\{food\} \pm c_3 \times Temp \times rand \times (X\_\{food\} - X\_i,j(t)) \quad (8)$$

Where $X_{i,j}$ denotes the position of a snake individual (male or female), X_{food} denotes the optimal position of a snake individual, rand is a random number in the range of [0,1], and c3 is a constant.

The principle of the snake optimization algorithm to optimize the random forest model is to find the optimal random forest model parameters by simulating the special mating behavior of snakes. In the snake optimization algorithm, the population is divided into two equal groups, males and females. Male snakes will fight with each other to attract the attention of females. Females have the right to decide whether to mate or not. If mating occurs, the female starts laying eggs in the nest or burrow and once the eggs appear, she leaves. The exploration and exploitation phases of the snake optimization algorithm are mainly affected by the temperature Temp and the amount of food Q. In the exploration phase, the snake will be in a state of searching for food, while in the exploitation phase, the snake will develop and optimize based on the location of food. By continuously updating and evolving the population, the snake optimization algorithm can eventually find the optimal random forest model parameters (see Fig. 1).

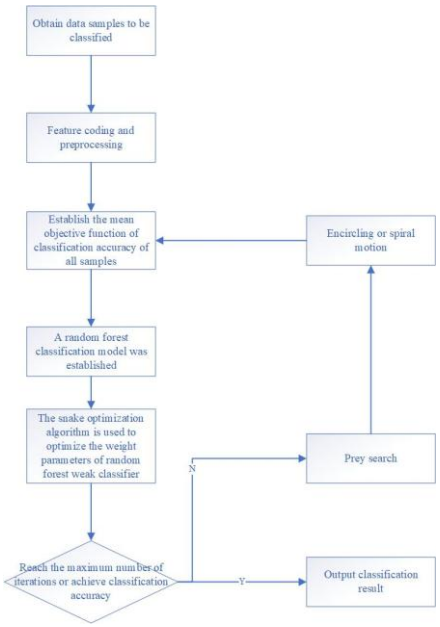


Figure 1: Flowchart of the snake optimization algorithm for optimizing the random forest framework

The Snake Optimization Algorithm (SO) improves model performance by optimizing important Random Forest (RF) parameters such as the number of trees, maximum depth, and minimum sample split. The process starts with population initialization, in which each snake signifies a unique set of RF parameters. SO creates various parameter sets during the exploration phase, ensuring a large search space, and then improves these configurations based on the mating tactic, enabling superior parameter sets to evolve. Fitness evaluation assesses model performance utilizing metrics such as accuracy and F1-score. Finally, during the selection and update phase, the best-performing parameter sets are retained and refined, guaranteeing that the RF model is optimally tuned for enhanced prediction accuracy and generalization.

3.4 Parameter tuning in snake optimization algorithm for random forest

The SO optimizes important Random Forest (RF) hyperparameters, like the number of estimators (trees), maximum tree depth, and minimum samples per split, by dynamically searching for the best configuration that optimizes model performance. First, SO creates a random population of hyperparameter sets, each indicating a possible RF configuration. The fitness function assesses each set using metrics such as accuracy, recall, and F1-score. The position update equation simulates snake evolution by balancing exploration (searching for novel hyperparameter regions) and exploitation (fine-tuning promising regions). Male individuals compete, while females choose the best candidates, guaranteeing diversity in hyperparameter selection. The iteration procedure continues until convergence, when the

optimum RF configuration is determined. This adaptive tuning method reduces overfitting and improves generalization across datasets. Pseudocode 1 shows SO-Based RF Hyperparameter Optimization.

Pseudocode 1: Snake optimization for random forest hyperparameter tuning

Initialize population (Snakes) with random values for:
- Number of estimators (n_estimators)
- Maximum tree depth (max_depth)
- Minimum samples per split (min_samples_split)

Set algorithm parameters:
- Population size (N), maximum iterations (T)
- Temperature (Temp), Food Quantity (Q)
- Exploration constant (c3)

Assess initial fitness for each snake utilizing:
Fitness = Performance Metric (for example Accuracy, F1-score)

FOR iteration t = 1 to T:
FOR each snake i:
- Update position utilizing:
$X_{i,j}(t+1) = X_{food} \pm c3 \times Temp \times rand \times (X_{food} - X_{i,j}(t))$
- Assess novel fitness score

Choose top-performing individuals (elite solutions)
Update population using mating tactic:
- Males fight for superior positions
- Females select best parameters and improve solutions

Adjust exploration vs. exploitation using temperature (Temp) decay

Check termination condition:
- If convergence is attained or max iterations reached, stop.

Return superior hyperparameter set (Optimum RF configuration)

This pseudocode guarantees reproducibility and shows how SO improves RF hyperparameters to improve prediction accuracy while maintaining computational effectiveness.

4 Empirical analysis

4.1 Data

According to the data of an e-commerce platform, 4374 sets of data were selected, and browsing records, purchase history, search keywords, click rate, dwell time, number of times of adding a shopping cart, user comments, and visit time were chosen as input indicators, and user conversion rate and purchase rate were chosen as output indicators (Bucklin R E, 2009). Among them, users' browsing records reflect their attention and interest in the products. This indicator can help the model understand users' shopping preferences and needs, to better predict their purchasing behavior. Users' purchase history records whether they have made purchases in the past, as well as the types and quantities of goods they have purchased and other information. This is valuable for predicting a user's future purchasing behavior, as it can provide important information about a user's purchasing ability and preferences. Users' search keywords on the platform can reveal their shopping needs and interests. Models can use these keywords to understand users' needs and predict what they are likely to buy. Click-through rate is the ratio of the number of times a user clicks on an item to the total number of items viewed, and it reflects the user's interest and preference for the item. Models can use this metric to predict what users are likely to buy. The amount of time a user spends on a page can provide information about their level of interest in the item. If users spend more time on the product page, they are more likely to purchase the item. The number of times a user adds an item to their shopping cart can reflect their level of interest in the item. Multiple additions to the cart may indicate a higher willingness to buy. User reviews provide their feedback on the item, which is valuable in understanding the user's level of satisfaction and possible shopping needs. Models can use this information to predict what users are likely to buy and their likely purchasing behavior. A user's online time can provide information about their shopping behavior and habits. If a user often shops between 7 pm and 10 pm, the model can use this information to predict what they are likely to buy during this time in the future. The reason for choosing user conversion rate and purchase rate as output metrics is that they are key metrics that directly measure how well an e-commerce platform is operating. Predicting these two metrics can help e-commerce platforms better formulate marketing strategies to improve revenue and operational efficiency.

In this paper, 4374 sets of data are selected, of which, 2624 sets of data are the training set, 1698 sets of data are the validation set, 52 sets of data are the test set, and the mean absolute error MAE, mean relative error MAPE, root mean square error MSE, root mean square error RMSE, and R2 of each set are calculated and analyzed by comparing them with other prediction models.

To ensure high-quality input data, numerous data preprocessing steps were used. Missing values were managed utilizing imputation methods like mean/mode imputation for numerical attributes (for example, dwell time, visit duration) and the most frequent category for categorical attributes (for example, search terms). Feature scaling was used with Min-Max Normalization to standardize numerical indicators such as click rate, shopping cart additions, and dwell time within a 0-1 range, avoiding models from being biased toward features with higher magnitudes. To guarantee predictive model compatibility, categorical encoding was executed utilizing one-hot encoding for nominal variables and label encoding for ordinal variables. The training-validation-test split (60%-38%-2%), while unconventional, is tactically designed: the large validation set (38%) guarantees resilient hyperparameter tuning and avoids overfitting, particularly for models that require extensive tuning, like ensemble learning. The small test set (2%) simulates practical e-commerce scenarios in which newly gathered, previously unseen data becomes available incrementally and acts as a final check for model generalization. This split structure prioritizes model optimization and validation while guaranteeing that the final performance evaluation is independent and realistic. The dataset size of 4,374 samples, while structured with pertinent features, is enough for training a random forest model, as the model's resilience in managing small to medium datasets reduces possible overfitting via ensemble learning.

4.2 Analysis steps

The experiments were carried out in a Windows 11 environment utilizing Python 3.9, with important libraries such as Scikit-learn (v1.2.2) for Random Forest, NumPy (v1.23) for numerical computations, and Matplotlib (v3.6) for visualizations. The Snake Optimization Algorithm (SOA) was executed with custom Python scripts that enabled parallel processing for effectiveness. The experiments were carried out on a workstation with Windows 11, an Intel Core i9-12900K processor, 64GB RAM, and an NVIDIA RTX 3090 GPU to ensure fast model training and hyperparameter tuning. The parameter search space for RF comprised tree depth (ranging from 5 to 50), number of estimators (50 to 500), minimum samples per split (2 to 20), and SOA-optimized feature selection thresholds. These computational settings allowed for comprehensive hyperparameter tuning, guaranteeing that the SOA-RF model was optimal for predicting e-commerce behavior.

Incorporating e-commerce user behavior prediction into the principle of snake optimization algorithm can be achieved by the following steps:

- collecting e-commerce user behavior data, including browsing records, purchase records, search records, etc.
- pre-processing and feature engineering of user behavior data to extract meaningful features, such as user browsing duration, purchase frequency, search keywords, etc.
- Use the snake optimization algorithm to select and optimize the features to find the optimal feature combination.
- Construct the e-commerce user behavior prediction model based on the optimal feature combination.
- Train and validate the model using historical data to evaluate the performance of the model.
- personalized recommendation and marketing for e-commerce users based on the prediction results of the model.

In this process, the snake optimization algorithm can be continuously explored and developed to find the optimal combination of features and model parameters, to improve the performance and accuracy of the e-commerce user behavior prediction model. At the same time, the e-commerce user behavior prediction model can also provide strong support and a basis for the personalized recommendation and marketing of the e-commerce platform, to achieve better user experience and commercial value.

4.3 Empirical analysis results

Table 1 shows a comparison table of the optimization results of the intelligent algorithms. The data in the table shows the performance metrics of different intelligent algorithms (RF, SA-RF, SSA-RF, SO-RF) on the training set, validation set, and test set. The SA-simulated annealing algorithm is an optimization algorithm based on a physical annealing process. It tries to explore more solution space by introducing a random perturbation to find the global optimal solution. When optimizing a random forest model, the SA algorithm can help to jump out of the local optimal solution and improve the prediction performance of the model.SSA Sparrow Search Algorithm is a heuristic search algorithm that finds the global optimal solution by simulating the flock search behavior of sparrows. The algorithm can maintain the diversity of the population during the search process, thus effectively avoiding falling into the local optimal solution. When optimizing the random forest model, the SSA algorithm can help explore more solution space and improve the prediction accuracy of the model. The snake optimization algorithm is an optimization algorithm based on biological snakes. It finds the global optimal solution by simulating the predation and regeneration behaviors of snakes. When optimizing the random forest model, the snake optimization algorithm can use the snake's movement characteristics to gradually improve the performance of the model and find better prediction results. The snake optimization algorithm has high search capability and optimality finding performance. It can effectively find the global optimal solution by simulating the predation and regeneration behavior of snakes. In contrast, the SA simulated annealing algorithm and SSA sparrow search algorithm may be more likely to fall into local optimal solutions. The snake optimization algorithm can make full use of the topology and information of the space during the search process, which helps to find the global optimal solution faster. The snake optimization algorithm is highly adaptive, and it can automatically adjust its parameters for different problems, thus better adapting to various complex data sets.

The study's goal is to predict user conversion and purchase rates using behavioral data. However, instead of displaying raw predicted values, the model's predictive performance is assessed utilizing Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE), Root Mean Square Error (RMSE), and Coefficient of Determination (R^2), which measure the difference between predicted and actual values.

The evaluation metrics MAE, MAPE, MSE, RMSE, and R^2 offer an extensive evaluation of model effectiveness. R^2 is prioritized as it assesses the extent to which independent variables account for variation in user conversion and purchase rates. A higher R^2 value denotes better predictive accuracy, which is important for e-commerce applications where comprehending user behavior patterns directly influences revenue and marketing tactics.

Compared to other models (RF, SA-RF, SSA-RF, and SO-RF), the SO-RF model consistently has the highest R^2 and the lowest MAE, MAPE, MSE, and RMSE, indicating superior generalization capacity. The MAE and RMSE values confirm lower absolute and squared errors, guaranteeing stable predictions, whereas MAPE emphasizes the relative error percentage, which is especially helpful when dealing with different scales of purchase behaviors. The statistical comparison shows that SO-RF's adaptive parameter tuning substantially decreases prediction errors, resulting in higher accuracy and resilience than other models.

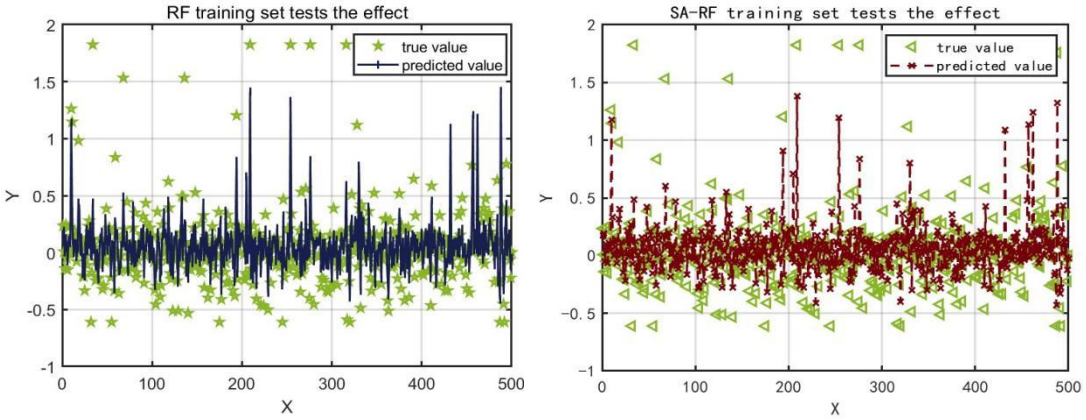
The findings contain an analysis of feature importance that employs SHAP values to improve interpretability. SHAP offers a detailed breakdown of how each feature impacts the model's predictions, allowing for more transparent decision-making. Visualizing SHAP values allows you to identify the most influential factors driving the model's performance, resulting in a better comprehension of how user behavior metrics contribute to prediction results. This information helps to refine marketing tactics and improve model dependability.

The random forest feature selection formula determines which features are important in impacting user conversion and purchase rates. Features with higher selection probabilities have an important effect on decision-making, emphasizing their relevance in

predicting user behavior. Important features like product interactions, browsing patterns, and engagement metrics have a direct impact on conversion rates because they reflect user interest and intent. Similarly, purchase-related features, such as prior transaction history and interaction frequency, help the model predict purchasing decisions. By prioritizing the most influential features, the model enhances predictive accuracy and ensures that only the most pertinent factors influence decision-making.
It can be seen from the data in the table:
As far as the MAE metric is concerned, SO-RF performs best on the training and test sets with 0.13431 and 0.31959 respectively. Random forest has an MAE of 0.17755 on the training set and 0.31029 on the test set.SA-RF and SSA-RF perform poorly on all the sets.
As far as the MAPE metrics are concerned, SO-RF performs best on the training and test sets with 0.90993 and 1.6652, respectively. Random forests have a MAPE of 1.2021 on the training set and 1.464 on the test set.SA-RF and SSA-RF perform poorly on all sets.
As far as the MSE metrics are concerned, SO-RF performs best on the training and validation sets with 0.058213 and 0.41112, respectively. Random Forest has an MSE of 0.10669 on the training set and 0.42372 on the validation set.SA-RF and SSA-RF perform poorly on all sets.
As far as RMSE metrics are concerned, SO-RF performs best on the training and validation sets with 0.24127 and 0.64118 respectively. Random Forest has an RMSE of 0.32663 on the training set and 0.65094 on the validation set.SA-RF and SSA-RF perform poorly on all sets.
As far as the R2 metric is concerned, SO-RF has the best performance on the training and test sets with 0.94325 and 0.96678, respectively. The random forest has an R2 of 0.88781 on the training set and 0.90983 on the test set.SA-RF and SSA-RF perform poorly on all sets.
In summary, the SO-RF algorithm outperforms the other three algorithms on the training, validation, and test sets in terms of each performance metric. This suggests that the SO-RF algorithm may be the best-performing of the four algorithms. Fig. 2, Fig. 3, and Fig. 4 show the training, validation, and test performance of various optimization algorithms by comparing true and predicted values, emphasizing their fitting accuracy, generalization capacity, and predictive capacity.

Table 1: Smart algorithm optimization results

model	set	**MAE**	**MAPE**	**MSE**	**RMSE**	**R2**
	train set	0.17755	1.2021	0.10669	0.32663	0.88781
RF	valid set	0.34081	2.2548	0.42372	0.65094	0.62007
	test set	0.31029	1.464	0.16939	0.41156	0.90983
	train set	0.15889	1.2909	0.41366	0.64316	0.6306
SA-RF	valid set	0.33947	2.1972	1.5851	1.259	0.68312
	test set	0.31685	1.7203	0.17198	0.4147	0.92049
	train set	0.15313	1.1137	0.078014	0.27931	0.92154
SSA-RF	valid set	0.3397	1.8564	0.42111	0.64893	0.6228
	test set	0.31175	1.5055	0.17316	0.41613	0.93524
	train set	0.13431	0.90993	0.058213	0.24127	0.94325
SO-RF	valid set	0.33994	2.0671	0.41112	0.64118	0.63333
	test set	0.31959	1.6652	0.17625	0.41983	0.96678



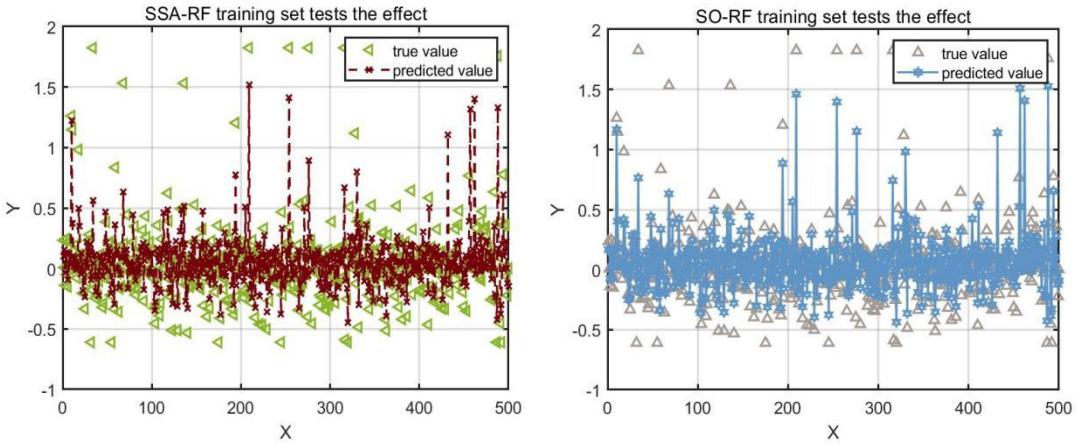


Figure 2: Effect of the training set on each optimization algorithm.

This figure compares true and predicted values during the training phase for various optimization algorithms. The X-axis indicates sample indices from 0 to 60, and the Y-axis indicates predicted values from -0.5 to 1. The legend distinguishes between true and predicted values and shows how well each algorithm fits the training data.

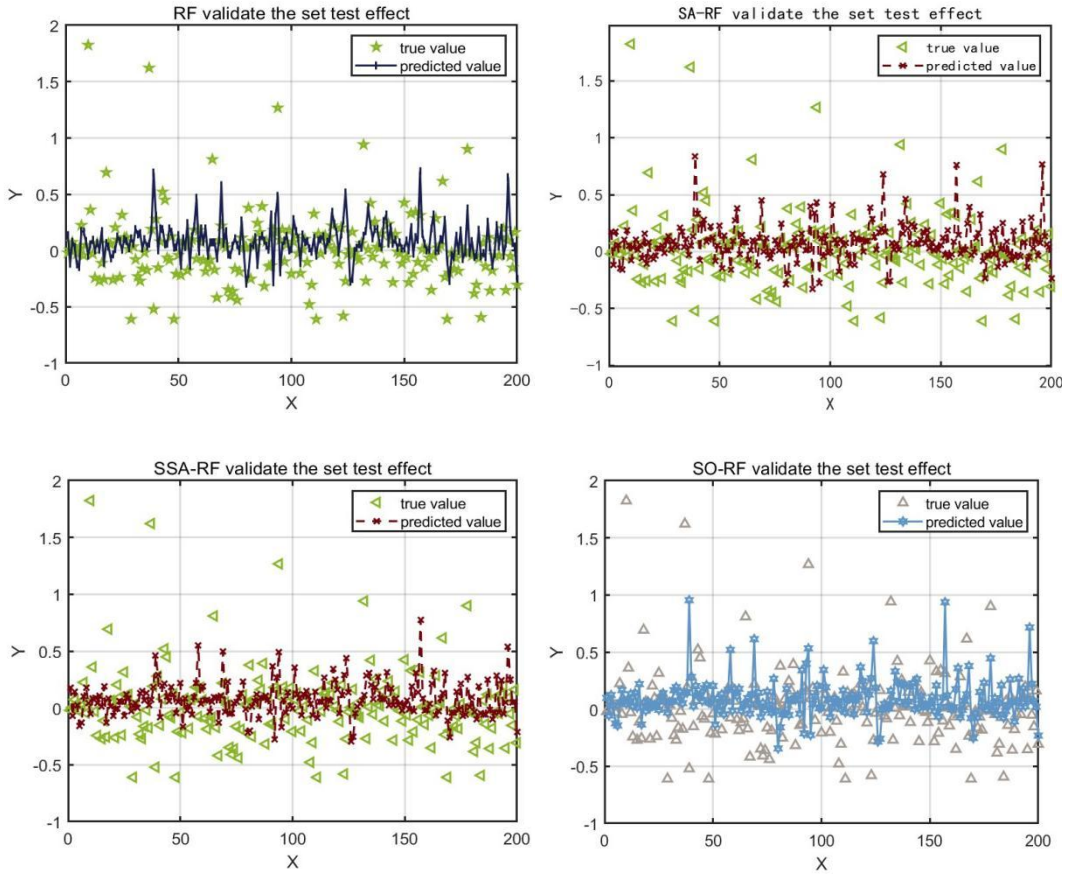


Figure 3: Effect of each optimization algorithm on the validation set.

This figure depicts the validation performance of various optimization algorithms by comparing true and predicted values. The X-axis shows validation samples (0-60), and the Y-axis shows predicted values (-0.5 to 1). The legend compares the actual and predicted values, emphasizing the accuracy of each algorithm in previously unseen data.

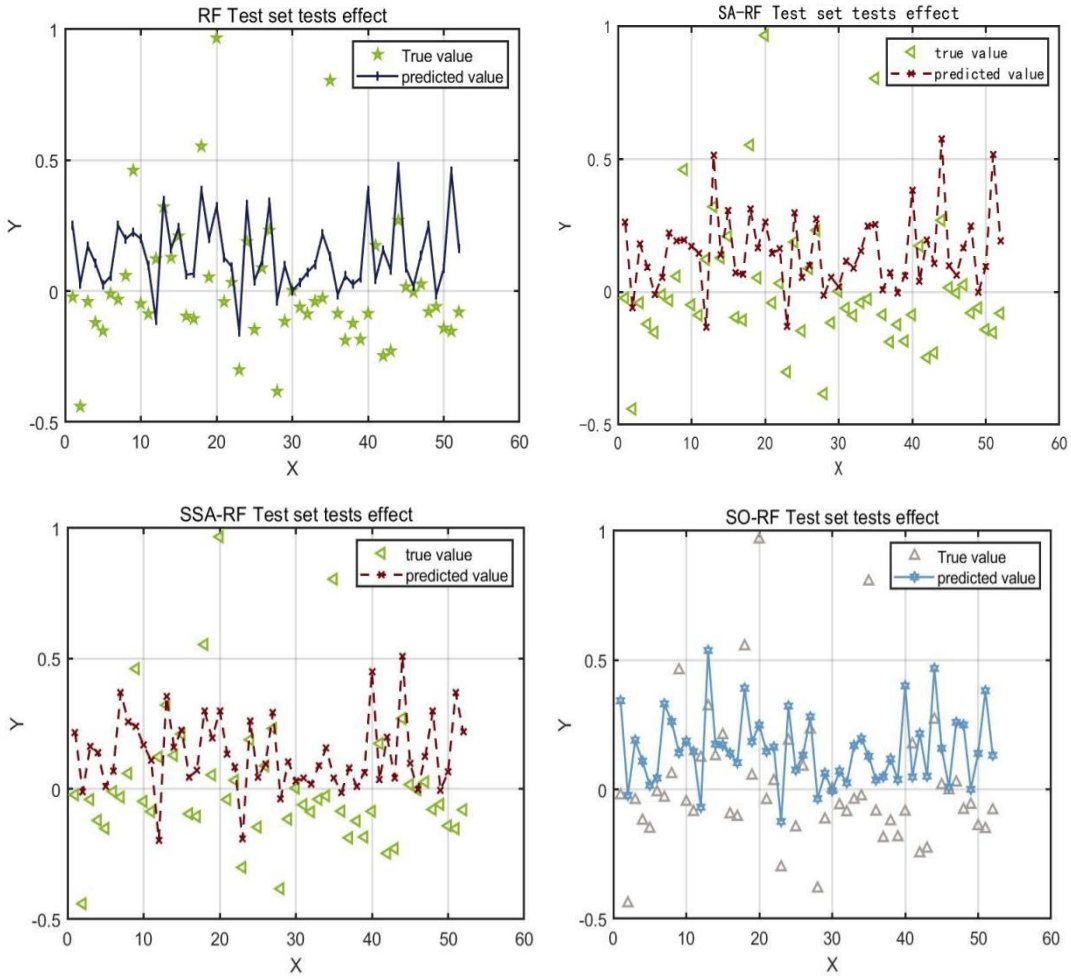


Figure 4: Effect of each optimization algorithm on the test set.

The true and predicted values for the test set are plotted in this figure to evaluate the generalization performance of various optimization algorithms. The X-axis ranges from 0 to 60, while the Y-axis is between -0.5 and 1. The legend distinguishes between actual and predicted values, demonstrating the predictive accuracy of each algorithm on independent test data

Table 1 compares the optimization findings for various intelligent algorithms, comprising RF, SA-RF, SSA-RF, and SO-RF, across the training, validation, and test datasets. The reported performance metrics show SO-RF's efficiency, with consistently higher accuracy and lower error rates than other approaches. The enhancement is due to adaptive parameter tuning in the SO framework, which improves model generalization. Notably, SO-RF surpasses SSA-RF and SA-RF in the validation and testing phases, demonstrating superior predictive stability on previously unseen data. These findings suggest that integrating Swarm Optimization (SO) substantially improves Random Forest (RF) performance,
rendering it a more reliable option for predicting e-commerce user behavior.

Table 2 shows that there is a certain positive correlation between user conversion rate and purchase rate. Users with higher conversion rates tend to have higher purchase rates, which indicates that for e-commerce platforms, improving user conversion rates can increase the frequency and amount of user purchases. There is a discrepancy between the prediction results of the random forest model and the actual values. The random forest model optimized by the snake optimization algorithm has better predictive performance compared to other optimization algorithms. This may be because the snake optimization algorithm has a high search capability and optimality finding performance, which can better find the optimal parameters and thus improve the predictive performance of the random forest model.

Table 2: Comparison of the real values of the test results of each optimization algorithm and the predicted values of each optimization algorithm

NO	True Value		Rf Predicted Value		Sa-Rf Predicted Value		Ssa-Rf Predicted Value		So-Rf Predicted Value	
	User conversion rate	Purchase rate	User conversion rate	Purchase rate	User conversion rate	Purchase rate	User conversion rate	Purchase rate	User conversion rate	Purchase rate
1	-0.0225	1.0147	0.2909	0.4674	0.2104	0.4396	0.2161	0.4230	0.2842	0.4406
2	-0.4403	0.9685	-0.0406	0.5890	-0.0625	0.5835	-0.0106	0.5430	-0.0310	0.5006
3	-0.0406	0.6402	0.1798	0.3665	0.2271	0.3878	0.1631	0.3984	0.1922	0.4094
4	-0.1205	1.1390	0.0752	0.4176	0.0544	0.3759	0.1394	0.4021	0.1057	0.3779
5	-0.1518	0.4502	0.0373	0.5595	0.0371	0.5636	0.0087	0.5355	0.0099	0.4843
6	-0.0106	1.2964	0.0672	0.4734	0.1373	0.4963	0.0676	0.5051	0.0652	0.4373
7	-0.0317	2.7359	0.2240	1.0652	0.2325	0.8913	0.3694	0.8463	0.1972	0.8792
8	0.0595	0.7566	0.1882	0.3533	0.1969	0.3583	0.2566	0.3438	0.2048	0.3499
9	0.4611	0.5291	0.2297	0.3250	0.1549	0.3624	0.2386	0.3570	0.2173	0.3442
10	-0.0480	0.5779	0.1686	0.3106	0.1586	0.3345	0.1676	0.3203	0.1621	0.3116
11	-0.0871	0.8783	0.1445	0.4766	0.1367	0.4145	0.1091	0.4211	0.1233	0.3738
12	0.1229	0.4823	-0.0811	0.3901	-0.0919	0.3721	-0.1959	0.3901	-0.1031	0.3707
13	0.3216	0.4894	0.4672	0.5777	0.5075	0.6163	0.3544	0.6002	0.4840	0.5967
14	0.1287	0.4450	0.1334	0.3193	0.1122	0.3307	0.1606	0.3384	0.1383	0.3148
15	0.2102	0.7606	0.2492	0.3775	0.2697	0.3656	0.2258	0.3790	0.2733	0.3391
16	-0.0957	1.0942	0.0844	0.4328	0.0985	0.3824	0.0456	0.4362	0.0879	0.3749

17	-0.1057	0.5227	0.0695	0.3601	0.0317	0.3862	0.0707	0.3955	0.1007	0.3749
18	0.5537	0.6968	0.3312	0.5894	0.4315	0.6397	0.2992	0.6267	0.3781	0.6103
19	0.0535	0.8909	0.2115	0.3873	0.2090	0.3958	0.1945	0.3896	0.1951	0.4303
20	0.9671	0.5130	0.2970	0.3839	0.2916	0.4065	0.2994	0.3971	0.2702	0.3998
21	-0.0407	0.3939	0.1611	0.3931	0.1482	0.4116	0.1337	0.4011	0.1442	0.4072
22	0.0327	0.8251	0.1389	0.3862	0.1085	0.4058	0.0828	0.3992	0.0990	0.3932
23	-0.3011	0.9533	-0.1317	0.3997	-0.1707	0.3774	-0.1927	0.3547	-0.1348	0.3892
24	0.1884	0.5071	0.2767	0.3250	0.2052	0.3643	0.2605	0.3704	0.2435	0.3464
25	-0.1464	1.2035	0.0487	0.4233	0.0009	0.4034	0.0444	0.3904	0.0436	0.3632
26	0.0883	0.8040	0.1311	0.3647	0.1032	0.3824	0.0948	0.3664	0.1244	0.3632
27	0.2319	0.5886	0.3327	0.3223	0.3424	0.3689	0.2935	0.3795	0.2817	0.3551
28	-0.3829	1.2094	-0.0078	0.4296	-0.0050	0.4060	-0.0375	0.4000	-0.0410	0.4167
29	-0.1158	0.9876	0.1244	0.3608	0.0864	0.3736	0.1040	0.3550	0.1083	0.3602
30	0.0029	0.7299	0.0363	0.3965	0.0403	0.3941	0.0308	0.3860	-0.0011	0.3937
31	-0.0609	0.8650	0.0617	0.4413	0.0638	0.4105	0.0412	0.3951	0.0690	0.3533
32	-0.0878	1.0012	0.0497	0.4104	0.0442	0.4349	0.0173	0.3944	0.0450	0.3503
33	-0.0398	1.0614	0.0861	0.3687	0.1017	0.3974	0.0879	0.3893	0.1115	0.3991
34	-0.0266	0.5723	0.2296	0.5592	0.2502	0.4868	0.1562	0.4695	0.2057	0.4621
35	0.8046	0.6362	0.1646	0.4555	0.2431	0.4479	0.0421	0.4355	0.1502	0.4131
36	-0.0854	0.7499	0.0313	0.3781	-0.0014	0.3837	-0.0138	0.3630	-0.0030	0.3715

37	-0.1877	0.7173	0.0129	0.4385	0.0163	0.4073	0.0789	0.4025	0.0527	0.4301
38	-0.1228	0.9489	0.0119	0.3668	0.0055	0.3546	0.0094	0.3395	0.0463	0.3213
39	-0.1845	0.8720	0.0527	0.4506	0.0156	0.4720	0.0627	0.4549	0.0334	0.4152
40	-0.0859	1.2428	0.3881	0.6393	0.4574	0.6163	0.4502	0.6569	0.3924	0.6273
41	0.1740	1.6601	0.0252	0.6414	-0.0083	0.7492	0.0355	0.6117	-0.0013	0.6266
42	-0.2468	0.3885	0.1957	0.3684	0.1947	0.3970	0.1972	0.3559	0.2046	0.3599
43	-0.2286	0.8800	0.0016	0.4250	0.0299	0.3959	0.0418	0.3795	0.0483	0.3873
44	0.2701	0.4075	0.5152	0.4336	0.4871	0.4412	0.5097	0.4369	0.4676	0.4155
45	0.0166	0.5846	0.0950	0.3972	0.1468	0.4002	0.0985	0.3749	0.1068	0.3757
46	-0.0034	0.7093	0.0088	0.4370	0.0169	0.4528	-0.0012	0.4282	0.0089	0.4404
47	0.0270	0.3643	0.1443	0.3906	0.1655	0.3904	0.1252	0.3750	0.1651	0.3662
48	-0.0788	0.7288	0.2593	0.3159	0.2237	0.3480	0.2974	0.3309	0.2413	0.3362
49	-0.0592	0.9161	-0.0188	0.4546	-0.0250	0.4485	-0.0039	0.3944	0.0079	0.4073
50	-0.1415	0.7889	0.0776	0.3818	0.0670	0.3910	0.0650	0.3787	0.1084	0.3636
51	-0.1528	0.6551	0.4245	0.3714	0.3984	0.3749	0.3690	0.3800	0.4221	0.3817
52	-0.0802	1.2139	0.1680	0.4575	0.1515	0.4219	0.2177	0.4219	0.1993	0.4210

The model's performance on unseen test data shows its capacity to generalize efficiently, with low MAE, MAPE, MSE, and RMSE values and a high R^2 score. The small test set (2% of the dataset) guarantees an objective assessment while simulating real-world e-commerce scenarios in which new data is constantly arriving. The findings indicate that the SO-RF model accurately captures intricate relationships in user behavior while reducing overfitting through adaptive parameter tuning. While the model generalizes well to novel data, its performance could be improved further using methods such as data augmentation or transfer learning, guaranteeing resilience across multiple e-commerce platforms.

The enhanced predictive performance of the SO-RF model has important practical implications for e-commerce platforms. Businesses can improve the accuracy of user behavior prediction to improve personalized suggestions, resulting in increased consumer engagement and conversion rates. More accurate demand prediction allows for more effective inventory management, decreasing overstocking and

stockouts, which has a direct influence on revenue growth. Furthermore, adaptive parameter tuning in SO improves model robustness, enabling platforms to dynamically adjust marketing tactics in response to changing customer tastes. This leads to more efficient targeted advertising, higher customer satisfaction, and increased brand loyalty, all of which contribute to a competitive benefit in the quickly evolving e-commerce environment.

5 Discussion

The empirical analysis shows that the proposed SO-RF model surpasses current state-of-the-art models for predicting e-commerce user behavior. SO-RF outperforms conventional Random Forest (RF) and optimization-enhanced variants like Simulated Annealing (SA-RF) and Sparrow Search Algorithm (SSA-RF) on all performance metrics, especially MAE, MAPE, MSE, RMSE, and R^2. The findings show that SO-RF consistently reduces prediction errors while increasing model dependability, especially in capturing complex user behavior patterns.

The SO algorithm has a significant advantage in that it allows for effective exploration and exploitation of the feature space through adaptive parameter tuning. Unlike SA, which may experience premature convergence because of its probabilistic cooling mechanism, and SSA, which mainly depends on swarm intelligence with limited adaptability, SO dynamically adjusts its search tactic in response to predatory and regenerative behaviors. This results in a more robust optimization process, enabling the RF model to strike a balanced trade-off between bias and variance. Additionally, SO's ability to use topological information improves feature selection, resulting in an optimal subset that enhances model interpretability and generalization.

However, despite its benefits, the SO-RF approach has some disadvantages. SO's computational complexity exceeds that of SA and SSA due to its iterative parameter adjustment and extensive search procedure. This could lead to longer training times, especially when dealing with large-scale e-commerce datasets. To address this, future research can look into hybrid optimization strategies that use early stopping mechanisms or parallel computing methods to improve efficiency.

Another critical consideration is dataset variability. The SO-RF model significantly improved prediction accuracy for the given dataset, but its efficacy may vary across e-commerce platforms with various user behavior patterns. Certain metrics, like MAPE and RMSE, may favor SO-RF because of its superior ability to reduce relative errors, but its efficacy should be validated on more diverse datasets.

To improve generalization, future research could combine deep learning architectures and SO-RF to capture nonlinear dependencies in user behavior. Furthermore, integrating real-time learning strategies could improve the model's adaptability to changing e-commerce settings. Overall, SO-RF represents a promising improvement in user behavior prediction, demonstrating high accuracy and robustness while emphasizing the requirement for additional optimizations to decrease computational costs and enhance scalability.

6 Conclusion

The Snake Optimization Algorithm efficiently improves the accuracy and performance of the Random Forest model through optimization. By simulating the special mating behavior of snakes, the snake optimization algorithm can find the optimal random forest model parameters, including the number of decision trees, the depth of decision trees, and feature selection. During the optimization process, the snake optimization algorithm automatically adjusts the values of the parameters according to the performance of the model to find the optimal model configuration.

Compared with traditional parameter optimization methods, such as grid search and random search, the snake optimization algorithm has higher efficiency and accuracy. By simulating the behavior of natural organisms, the snake optimization algorithm has good optimization searching ability and fast convergence and can find the optimal random forest model parameters in a shorter time. In addition, the snake optimization algorithm can effectively control the overfitting problem of the random forest model, to improve the generalization ability and robustness of the model.

In practical applications, the snake optimization algorithm can be widely used in a variety of practical problems, such as function optimization, machine learning, deep learning, and so on. By integrating into practical application scenarios in other fields such as e-commerce user behavior prediction, snake optimization algorithms can provide effective support and a basis for solving complex problems, to achieve better application value and commercial benefits.

In general, through the analysis and discussion of the snake optimization algorithm in optimizing the random forest model, the following conclusions can be drawn: the snake optimization algorithm has high efficiency and accuracy and can find the optimal parameters of the random forest model in a shorter period. The snake optimization algorithm can be applied to various practical problems by simulating the behavior of natural organisms with good optimization-seeking ability and fast convergence. The snake optimization algorithm can effectively control the overfitting problem of the random forest model, to improve the generalization ability and robustness of the model. By integrating into practical application scenarios in other fields such as e-commerce user behavior prediction, the snake optimization algorithm can provide effective support and a basis for solving complex problems, to achieve better application value and commercial benefits. The application of snake

optimization algorithms in other fields can be further explored in the future to give full play to its advantages and potential.

6.1 Limitations & future work

Despite its efficacy, the Snake Optimization Algorithm has a few limitations. For starters, its performance is sensitive to hyperparameter settings, necessitating precise tuning to attain the best results. Second, when dealing with high-dimensional datasets, the algorithm may have a slower convergence rate, potentially increasing computational costs. Furthermore, while it improves generalization, its capacity to prevent overfitting may differ between datasets and problem domains.

In the future, integrating hybrid optimization techniques, such as Snake Optimization with Bayesian Optimization or Genetic Algorithms, could improve parameter tuning effectiveness. Furthermore, combining deep learning models with the optimized Random Forest may improve predictive accuracy for complex e-commerce behavior patterns. Finally, broadening the study to include real-time user behavior prediction and adaptive marketing tactics would increase its practical utility in dynamic e-commerce environments.

References

[1] Amaratunga D, Cabrera J, Lee Y S., 2008, Enriched random forests[J]. *Bioinformatics*, 24(18):2010-2014. https://doi.org/10.1093/bioinformatics/btn356

[2] Ao Y, Li H, Zhu L, et al., 2019, The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling[J]. *Journal of Petroleum Science and Engineering*, 174: 776-789. https://doi.org/10.1016/j.petrol.2018.11.067

[3] Bin J, Ai F F, Fan W, et al., 2016, A modified random forest approach to improve multi-class classification performance of tobacco leaf grades coupled with NIR spectroscopy[J*]. RSC advances,* 6(36): 30353-30361. https://doi.org/10.1039/c5ra25052h

[4] Blazevic V, Lievens A., 2008, Managing innovation through customer coproduced knowledge in electronic services: An exploratory study[J]. *Journal of the academy of marketing science,* 36: 138-151. https://doi.org/10.1007/s11747-007-0064-y

[5] Bucklin R E, Sismeiro C., 2009, Click here for Internet insight: Advances in clickstream data analysis in marketing[J]. *Journal of Interactive Marketing*, 23(1): 35-48. https://doi.org/10.1016/j.intmar.2008.10.004

[6] Chang K, Jackson J, Grover V., 2003, E-commerce and corporate strategy: an executive perspective[J]. *Information & Management*, 40(7): 663-675. https://doi.org/10.1016/s0378-7206(02)00095-2

[7] Cheng X, He H., 2024, Sep 25, Enhancing Product Modelling Process Design and Visual Performance Through Random Forest Optimization. *Informatica,* 48(14). https://doi.org/10.31449/inf.v48i14.5800

[8] Cutler D R, Edwards Jr T C, Beard K H, et al., 2007, Random forests for classification in ecology[J]. *Ecology,* 88(11): 2783-2792. https://doi.org/10.1890/07-0539.1

[9] Fan S, Lau R Y K, Zhao J L., 2015, Demystifying big data analytics for business intelligence through the lens of marketing mix[J]. *Big Data Research,* 2(1): 28-32. https://doi.org/10.1016/j.bdr.2015.02.006

[10] Fawagreh K, Gaber M M, Elyan E., 2014, Random forests: from early developments to recent advancements[J]. *Systems Science & Control Engineering: An Open Access Journal,* 2(1): 602-609. https://doi.org/10.1080/21642583.2014.956265

[11] Fu H, Shi H, Xu Y, et al., 2022, Research on Gas Outburst Prediction Model Based on Multiple Strategy Fusion Improved Snake Optimization Algorithm with Temporal Convolutional Network[J]. *IEEE Access,* 10: 117973-117984. https://doi.org/10.1109/access.2022.3220765

[12] Guo Y, Yin C, Li M, et al., 2018, Mobile e-commerce recommendation system based on multi-source information fusion for sustainable e-business[J]. *Sustainability,* 10(1): 147. https://doi.org/10.3390/su10010147

[13] Hashim F A, Hussien A G., 2022, Snake Optimizer: A novel meta-heuristic optimization algorithm[J]. *Knowledge-Based Systems,* 242: 108320. https://doi.org/10.1016/j.knosys.2022.108320

[14] Khrais L T., 2020, Role of artificial intelligence in shaping consumer demand in E-commerce[J]. *Future Internet,* 12(12): 226. https://doi.org/10.3390/fi12120226

[15] Li T, Zhou M. 2016, ECG classification using wavelet packet entropy and random forests[J]. *Entropy*, 18(8): 285. https://doi.org/10.3390/e18080285

[16] Lin W, Wu Z, Lin L, et al., 2017, An ensemble random forest algorithm for insurance big data analysis[J]. *Ieee access,* 5: 16568-16575. https://doi.org/10.1109/access.2017.2738069

[17] Naghibi S A, Pourghasemi H R, Dixon B., 2016, GIS-based groundwater potential mapping using boosted regression tree, classification and regression tree, and random forest machine learning models in Iran[J]. *Environmental monitoring and assessment,* 188: 1-27. https://doi.org/10.1007/s10661-015-5049-6

[18] Niu Z, Wu J, Liu X, et al., 2021, Understanding

energy demand behaviors through spatio-temporal smart meter data analysis[J]. *Energy,* 226: 120493. https://doi.org/10.1016/j.energy.2021.120493

[19] Poggi N, Muthusamy V, Carrera D, et al., 2013. Business process mining from e-commerce web logs[C]//Business Process Management: 11th *International Conference, BPM 2013, Beijing, China,* August 26-30, *Proceedings. Springer Berlin Heidelberg,* 2013: 65-80. https://doi.org/10.1007/978-3-642-40176-3_7

[20] Ren S, Cao X, Wei Y, et al., 2015: Global refinement of random forest[C]//*Proceedings of the IEEE conference on computer vision and pattern recognition*. 723-730. https://doi.org/10.1109/cvpr.2014.218

[21] Segal M, Xiao Y., 2011, Multivariate random forests[J]. *Wiley Interdisciplinary Reviews: Data mining and knowledge discovery*, 1(1): 80-87. https://doi.org/10.1002/widm.12

[22] Shi K, Qiao Y, Zhao W, et al., 2018, An improved random forest model of short-term wind-power forecasting to enhance accuracy, efficiency, and robustness[J]. *Wind energy,* 21(12): 1383-1394. https://doi.org/10.1002/we.2261

[23] To M L, Ngai E W T., 2006, Predicting the organizational adoption of B2C e-commerce: an empirical study[J]. *Industrial Management & Data Systems,* 106(8): 1133-1147. https://doi.org/10.1108/02635570610710791

[24] Verikas A, Gelzinis A, Bacauskiene M., 2011, Mining data with random forests: A survey and results of new tests[J]. *Pattern recognition,* 44(2): 330-349. https://doi.org/10.1016/j.patcog.2010.08.011

[25] Wakil K, Alyari F, Ghasvari M, et al. 2020, A new model for assessing the role of customer behavior history, product classification, and prices on the success of the recommender systems in e-commerce[J]. *Kybernetes,* 49(5): 1325-1346. https://doi.org/10.1108/k-03-2019-0199

[26] Wu Z, Shen S, Zhou H, et al., 2021, An effective approach for the protection of user commodity viewing privacy in e-commerce website[J]. *Knowledge-Based Systems,* 220: 106952. https://doi.org/10.1016/j.knosys.2021.106952

[27] Xiahou X, Harada Y., 2022, B2C E-commerce customer churn prediction based on K-means and SVM[J]. *Journal of Theoretical and Applied Electronic Commerce Research,* 17(2): 458-475. https://doi.org/10.3390/jtaer17020024

[28] Xie C, Xiao X, Hassan D K., 2020, Data mining and application of social e-commerce users based on big data of internet of things[J]. *Journal of Intelligent & Fuzzy Systems,* 39(4): 5171-5181. https://doi.org/10.3233/jifs-189002

[29] Yan C, Razmjooy N., 2023, Optimal lung cancer detection based on CNN optimized and improved Snake optimization algorithm[J]. *Biomedical Signal Processing and Control,* 86: 105319. https://doi.org/10.1016/j.bspc.2023.105319

[30] Yan P, Zhou Y., 2024 Jun 10, Application of Recommendation Algorithm Based on Matrix Dimensionality Reduction Model in Network Information Analysis Model. *Informatica,* 48(9). https://doi.org/10.31449/inf.v48i9.5969

[31] Yuan Z., 2024 Sep 26, Consumer behavior prediction and enterprise precision marketing strategy based on deep learning. Informatica, 48(15). https://doi.org/10.31449/inf.v48i15.6260

[32] Zhang B, Wang L, Li Y., 2021, Precision marketing method of E-commerce platform based on clustering algorithm[J]. *Complexity,* 2021: 1-10. https://doi.org/10.1155/2021/5538677

[33] Zhang W, Wu C, Li Y, et al., 2021, Assessment of pile drivability using random forest regression and multivariate adaptive regression splines[J]. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards,* 15(1): 27-40. https://doi.org/10.1080/17499518.2019.1674340

[34] Zheng W, Pang S, Liu N, et al., 2023, A Compact Snake Optimization Algorithm in the Application of WKNN Fingerprint Localization[J]. *Sensors,* 23(14): 6282. https://doi.org/10.3390/s23146282