
A User Behavior Prediction Method for Web Applications Based on Deep Forest

Chang-Sheng Ma^{1,*}, Xiang-Ran Du², Jing Lou¹
and Ming-Qian Wang¹

¹*School of Information Engineering, Changzhou Vocational Institute of Mechatronic Technology, Jiangsu 213164, China*

²*Department of Information Engineering, Tianjin Maritime College, Tianjin 300457, China*

E-mail: machangsheng@czimt.edu.cn

**Corresponding Author*

Received 26 September 2024; Accepted 23 November 2024

Abstract

To increase the sales of agricultural products in e-commerce, understanding customer preferences is essential. In agricultural web applications, data mining techniques can help farmers analyze customer behavior patterns and identify preferences, thus optimizing product design or offering more precise personalized services, which, in turn, can enhance farmers' decision-making in agricultural production. This study proposes a web application user behavior prediction method based on deep forest, which addresses the issue of traditional learning methods requiring a large number of hyperparameter settings. Analysis results show that the Mondrian deep forest model has an accuracy of 95.42% and a running time of 55 s. The accuracy and efficiency of the Mondrian deep forest model are higher than for other models, and the proposed model can improve the accuracy of predicting user behavior in web applications. The effectiveness of the algorithm has been validated through practical testing.

Keywords: Deep forest (DF), web application, user behavior, prediction.

Journal of Web Engineering, Vol. 24_1, 39–56.

doi: 10.13052/jwe1540-9589.2412

© 2025 River Publishers

1 Introduction

Agriculture holds a vital position in China's national economy, and leveraging informatization to drive industrialization is the only way to achieve leapfrog development in the agricultural sector. In the era of the "network economy," agricultural enterprises are shifting from a "product-centered" to a "customer-centered" approach in management. Agricultural web applications can be utilized by small and medium-sized agricultural product enterprises, enabling farmers to conduct remote online ordering and sales management. By sharing and transmitting sales information up and down the chain, they can quickly handle various tasks, control and analyze sales conditions in real time, guide production, and optimize resources like inventory. Farmers can use the latest web applications to efficiently collect, publish, store, process, and communicate customer information, expanding their agricultural product sales at a low cost. Sales are a critical link in the production and operation of farms, and the rapid development of web application service technologies provides more space and opportunities for agricultural product sales management in the new economy era. By harnessing the advantages of web applications, data can be transmitted quickly and conveniently, replacing the traditional method of manual record-keeping via telephone communication. This significantly improves work quality and efficiency.

The browsing and operational data generated by customers while using agricultural web applications contain valuable insights into customer preferences. Extracting useful information from this vast amount of customer and product data has become a key focus in e-commerce research. Predicting customer purchasing behavior is crucial for enhancing customer experience and promoting the development of agricultural e-commerce, offering both academic and practical value. While agricultural web applications provide convenience to customers in purchasing agricultural products, the rich variety of product categories and diverse marketing strategies often make it difficult for customers to make choices. Therefore, utilizing advanced techniques to deeply analyze and understand customer purchasing behaviors has become a critical research topic in both industry and academia. These insights provide reliable technical support for customer decision-making and agricultural enterprise operations, significantly improving the customer experience and fostering a mutually beneficial relationship between farmers and customers. Agricultural web applications gather customer consumption data and behavioral patterns. Effectively utilizing this data to extract valuable insights and predict future customer behavior based on historical information can greatly

enhance the precision of e-commerce product recommendations and improve the overall customer experience.

In recent years, many scholars have conducted extensive research on the issue of web application customer behavior prediction and have proposed numerous effective algorithms. Silahtaroglu G. and Donertasli H. introduced a prediction method based on decision trees and neural networks to forecast users' purchasing behavior for items in their shopping carts [1]. Wen Wen et al, proposed a dynamic behavior prediction method based on embedded learning, which offers high prediction accuracy, but the model's long training time hinders its practical application [2]. Hu Xiaoli et al., developed a prediction model based on ensemble learning, balancing the sample dataset and improving the model's training efficiency [3].

In recent years, massive amounts of user behavior data have been generated across various service platforms. By deeply mining this data, it is possible to uncover users' shopping habits and preferences. Leveraging users' behavioral data during the purchasing process for recommendation has become a viable method, drawing attention from several researchers. With the development of deep learning, researchers have attempted to use deep learning techniques to address the problem of predicting customer behavior in web applications. Strub et al. applied two stacked denoising autoencoders (SDAE), taking product ratings and user features as inputs to learn latent representations of users and items. These representations were then used to predict missing ratings. This method effectively solves the data sparsity issue in collaborative filtering, but because the method zeroes out missing values in the rating matrix to reduce network connections during training, it overlooks unrated information [4]. Xu et al. used the DSSM model to study tag-aware recommendation problems. They defined user input features using tag data from both users and products, then learned the latent representations of users and items. By calculating the similarity between these representations, they were able to make recommendations, addressing the challenge of user feature extraction in traditional content-based recommendation systems [5]. Wei et al., proposed a hybrid recommendation algorithm that combines a stacked denoising autoencoder (SDAE) with the collaborative filtering method TimeSVD++. This algorithm learns latent representations of products from auxiliary information using SDAE and fits the rating matrix between users and products using TimeSVD++, effectively addressing the difficulty of representing auxiliary data in hybrid recommendations [6]. While these three deep learning methods have successfully solved the problems faced by traditional recommendation systems, deep learning models are often complex,

require lengthy training times, and function as a “black box,” which raises concerns about the lack of interpretability in recommendation systems [7].

In recent years, researchers have considered applying users’ actions (clicking, browsing, adding to cart, bookmarking, purchasing) on some shopping websites to recommendation systems, aiming to explore users’ shopping habits and behavioral preferences based on their historical actions. Zeng Xianyu et al. proposed a method based on user action sequences and choice models, using an action sequence utility function to infer the optimal substitute products for users during the purchase cycle. They established a latent factor-based choice model for purchased products and optimal substitutes, thereby deriving users’ purchasing preferences [8]. Ding Zhe et al. proposed a recommendation model based on users’ browsing behavior, which can predict mobile users’ browsing behaviors on the internet, and, based on these predictions, recommend content to them. Although these two models effectively address issues like data sparsity and item cold start, they only consider a single user action, such as clicking (browsing). In real business scenarios, user behaviors include multiple actions such as clicking, browsing, bookmarking, adding to cart, and purchasing. Mining the overall behavior generated during the purchasing process will enhance the accuracy of recommendations [9].

In summary, these technologies are mostly based on the relationship between users and products for recommendations. Even when user behavior is considered, it only involves one type of action, without accounting for users’ overall actions. This paper addresses the traditional recommendation system’s problems with data sparsity, long training times for deep learning models, model complexity, the need for large-scale training data, and the inability to fully consider users’ overall actions in the recommendation process. DF is an effective means of predicting customer behavior and Dube et al. [10] propose a model based on DF for predicting customer purchasing behavior in agricultural product web applications using an ensemble learning approach. This model significantly improves prediction performance. DF is a tree-based learning model that is easier to analyze theoretically compared to deep learning recommendation models and requires far fewer parameters, making parameter tuning simpler, and therefore DF can improve the efficiency of prediction [11]. The parameter settings for this model are highly robust and, in most cases, using default values yields good results for agricultural data. Additionally, the model trains quickly and has high accuracy. Therefore, this paper utilizes the DF method, combined with

customer behavior data from agricultural product web applications, to build a recommendation model.

2 Overview of Agricultural Product Web Applications

The agricultural product web application is designed for product sales management in the agricultural product industry, allowing for information sharing with enterprise information management systems. This means that the system must manage agricultural product sales while also timely reflecting the status of the enterprise's production and operations. By using data to reflect sales prospects, the system provides support for production, operational, and sales decision-making. The main functions of the agricultural product web application are as follows. (1) Product purchase function: Users can order their preferred products by placing an order. (2) Product browsing: This includes information such as product prices, descriptions, and images. (3) Back-end management: This includes user management, product management, and site information management, therefore a Web application is an effective means of managing agricultural products [12].

The application uses ASP technology. When users access it, the Web server first calls the .asp file and reads its entire content. Using specialty products as an example, the main function is to manage all agricultural and subsidiary products of agricultural enterprises, including product sales, as well as adding, deleting, and modifying products. For product sales, it mainly involves two stages: customer order placement and administrator processing of the order. Customers can place orders for products based on the information provided on the webpage, while administrators process the order based on the information it contains. Customers can check the status of their orders to see if they have been processed and which ones remain pending. The addition and modification of products are entirely managed by administrators. Similarly, there are two stages: one is adding new product names, quantities, prices, and descriptions; the other is modifying information for existing products. This ensures that when new products are introduced, they can be published in a timely manner and managed efficiently, allowing for real-time system updates. The agricultural product web application enables the integration and interaction of the sales system, suppliers, customers, and different branches of the enterprise, applying security authentication mechanisms to online sales of agricultural products. The functional framework of the web application is shown in Figure 1.

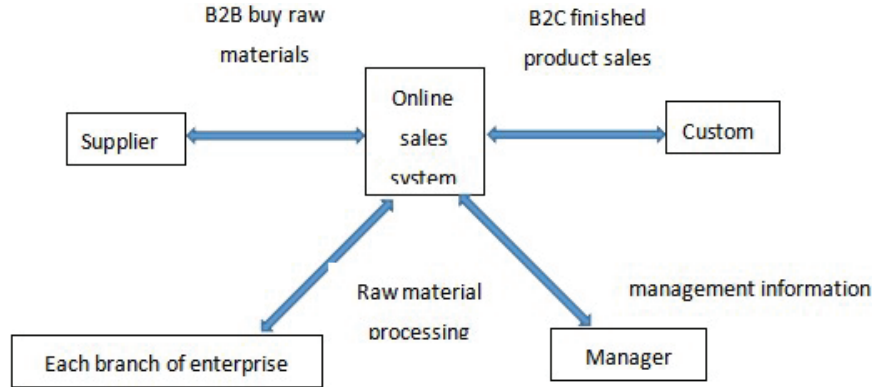


Figure 1 The functional framework of the web application.

3 Basic Theory of the Deep Forest Algorithm

The deep forest (DF) algorithm is a non-neural network deep model, and geForest is the first DF model. geForest has a cascaded forest structure, where each layer in the cascade is composed of multiple decision trees, including random forests and completely random forests. Each decision tree in the forest outputs its estimated class distribution, forming a class probability vector. This class probability vector, augmented with the original input features, is then fed into the next layer. As new layers are added to the cascade, cross-validation is used to estimate the overall performance of the cascade. When the desired performance is not achieved, the training process is terminated. Thus, the DF algorithm can automatically determine the number of cascade layers, meaning that the complexity of the model is automatically determined based on the data. The DF algorithm adopts an additive model and provides a new perspective for DF from the margin theory viewpoint, DF can obtain a better prediction effect [13].

Currently, extending decision tree forests to incremental/online learning has shown promising results. Most existing online decision tree forests need to maintain and update the candidate split list for each leaf node and their corresponding split quality scores, which leads to significant time and space overhead. To avoid this drawback, the Mondrian forest (MF) can be used. The MF differs significantly from other decision trees in its splitting process as it has some advantages in decision analysis [14]. The split selection in MF is independent of sample labels, and compared to other online random forests, MF has faster incremental updates and higher prediction accuracy.

With the same amount of training data, MF achieves better accuracy. For each node in the MF, a splitting dimension and splitting point are randomly sampled based on the range of the data in each dimension. Each node in the MF is also associated with a split time. This splitting mechanism and split time mechanism enable the MF to update efficiently. When a new training sample appears, the MF can choose one of three options based on the relative position of the new sample to the existing data at the node: (1) Introduce a higher-level split above the current one; (2) update the current split's range to include the new training sample; (3) split the current leaf node into two child nodes. This means that the MF can modify the structure of the entire tree, whereas other incremental random forests can only update leaf nodes. For a test sample, an MF outputs the predictive distribution over the labels. A Mondrian forest is an ensemble of multiple independent MFs, and its output is the average of the predictions from each tree.

A Mondrian DF integrates incremental learning capabilities into the structure of the cascaded forest. A Mondrian DF has a cascaded forest structure, where each layer of the cascade contains multiple Mondrian forests. The input to each Mondrian forest is the feature information processed by the previous cascade layer, and the output is the feature information processed by the current layer, which is then passed to the next layer [15].

Let Ω be the joint distribution over $\alpha \times \beta$, where α is the M -dimensional sample space, and $\beta = \{1, 2, \dots, D\}$ is the label space. Both training and testing samples are drawn independently and identically from this distribution. Let $\vec{g} = \{g_1, g_2, \dots, g_T\}$ and $\vec{k} = \{k_1, k_2, \dots, k_T\}$, where g_t represents the Mondrian forest at the t layer, and k_t represents the cascaded result of the Mondrian forest at the t layer. Therefore, a Mondrian DF can be formalized as (\vec{g}, \vec{k}) .

At the t layer, $k_t: \alpha \rightarrow \lambda$ is defined as [16]:

$$k_t(x) = \begin{cases} g_1(x) & t = 1 \\ g_1([x, g_{t-1}(x), \varsigma]) & t > 1 \end{cases} \quad (1)$$

Assuming that each layer contains only one Mondrian forest, then the output at the t th layer, k_t , is a class probability vector. The input to k_t is the original features X , and the input to each subsequent layer k_t is a concatenated vector of the original features X and the transformed features from the output of the previous layer. These are referred to as augmented features. When each layer contains multiple Mondrian forests, the multiple class probability vectors are concatenated together to

form the augmented features. α represents the adaptive factor, which adjusts the weights between the original features and the augmented features.

Each pair (X, α) defines a Mondrian DF model [17]:

$$s(x) = \arg \max_{d \in \{1, 2, \dots, D\}} [k_T(\vec{x})]_D \quad (2)$$

In a decision tree, each internal node corresponds to a split (μ_j, τ_j) , where μ_j represents the splitting dimension, and τ_j represents d the splitting point along that dimension. Let $U_l(j)$ and $U_r(j)$ represent the two child nodes. A split is defined as:

$$R_l(j) = \{x | x_{\mu_j} \leq \tau_j\} \quad (3)$$

$$R_r(j) = \{x | x_{\mu_j} > \tau_j\} \quad (4)$$

For node j , let ξ_{jd} and σ_{jd} represent the minimum and maximum values of the training data along dimension d at the node, and calculate the data range along each dimension $\gamma_{jd} = \sigma_{jd} - \xi_{jd}$. In the Mondrian tree, the splitting dimension is sampled with a probability proportional to the range γ_{jd} of the data along each dimension within the node [18].

$$\Pr(\mu_j = d) = \frac{\gamma_{jd}}{\sum_{d \in \{1, 2, \dots, D\}} \gamma_{jd}} \quad (5)$$

In the formula, \Pr represents the probability.

The adaptive factor ς is used to adjust the weights of augmented features and original input features during the random sampling process of selecting the splitting feature. Assuming the dimensionality of the augmented features is D' , and the total dimensionality after concatenating with the original features is $D + D'$, the probability of selecting an original input feature as the splitting feature is [19]:

$$\Pr(\mu_j = d) = \frac{\gamma_{jd}}{\sum_{d \in \{1, 2, \dots, D\}} \gamma_{jd} + \varsigma \sum_{d=D+1}^{D+D'} \gamma_{jd}}, \quad d \in \{1, 2, \dots, D\} \quad (6)$$

The probability of selecting an augmented feature is:

$$\Pr(\mu_j = d) = \frac{\varsigma \gamma_{jd}}{\sum_{d \in \{1, 2, \dots, D\}} \gamma_{jd} + \varsigma \sum_{d=D+1}^{D+D'} \gamma_{jd}}, \quad d \in \{D + 1, D + 2, \dots, D + D'\} \quad (7)$$

The adaptive factor ς is a tunable parameter, and there are multiple ways to set it. A simple balancing strategy is adopted, where for each dataset, a fixed adaptive factor $\varsigma = \frac{D}{D'}$ is set. Therefore:

$$\Pr(\mu_j \in \{1, 2, \dots, D\}) = \Pr(\mu_j \in \{D + 1, D + 2, \dots, D + D'\}) \quad (8)$$

In the incremental training setting, data arrives in batches over time. The goal is to update the model promptly with newly obtained training data, allowing the model to fully utilize existing training data to achieve the best possible predictive performance. For a Mondrian DF, by updating the Mondrian forests layer by layer, the entire cascade structure can be updated. Additionally, with dynamic adjustments to the number of effective layers, an incremental Mondrian DF can be obtained, which gradually increases model complexity as more training data is received, thereby improving performance.

Let Ψ represent the algorithm for generating a Mondrian forest g_t in the cascade structure. Assume the training data is divided into F small batches, denoted as $\{H^f\}_{f=1}^F$, and g_t^f represents the Mondrian forest in layer t after being updated with the f th batch of training data. When a new batch of training data H^f is received, the model in layer t , g_t^{f-1} , will be updated to g_t^f , according to the following update formula [20]:

$$g_t^f \leftarrow \begin{cases} \Psi(g_t^{f-1}, H^f) & t = 1 \\ \Psi(g_t^{f-1}, H^f, \varsigma) & t > 1 \end{cases} \quad (9)$$

When receiving the first batch of training data H^1 , a Mondrian DF is trained from scratch. Assuming the estimated optimal number of layers is C and considering that the complexity of the model increases as more training data is received, after layer C , additional C^+ layers are trained, but they are temporarily excluded from prediction and belong to inactive cascade layers. At this point, the total number of cascade layers is $C' = C + C^+$. When new training data H^f ($f > 1$) is received, the Mondrian forests are updated layer by layer. During the update process, all C' layers are updated. Meanwhile, the trend of cross-validation accuracy for the current batch of training samples across layers can be used to decide whether to activate more cascade layers for prediction.

4 Web Application Customer Behavior Prediction Model

A dataset was created by collecting sales data from an agricultural product web application. This dataset accurately reflects customers' historical

behaviors, and its broad coverage and reliability lay a solid foundation for algorithm validation. The dataset includes a customer information table, an agricultural product information table, a customer behavior table, a customer order table, and a review data table. Some fields have been processed for privacy protection. The dataset contains behavior data for over 100,000 customers and more than 5000 products.

First, eight tables from the dataset were merged into one table, from which 142 specific features were extracted, including basic features, quantitative features, time-series features, and combined features. After feature extraction, a feature matrix of 9500 rows by 142 columns was generated. However, statistical analysis showed that 81.93% of the dataset represents purchasing behavior, while only 18.07% represents non-purchasing behavior, indicating a significant imbalance in the dataset. Therefore, balancing the dataset is necessary to ensure that the predicted classes are more evenly distributed. After balancing, a feature matrix of 52,934 rows by 142 columns was obtained as the input for the DF model.

The Mondrian forest obtains probability feature vectors by processing the original feature matrix through multi-granularity scans, and then achieves the final prediction result through the layered learning of cascade forests. Based on the 142-dimensional feature vectors obtained from preprocessing, sliding windows of lengths $n/4$, $n/2$ and $3n/4$ were designed to extract features, with a sliding step size of 1. The multi-granularity scanning phase employs two Mondrian forests, each with 800 decision trees, and the sliding window dimensions are 60, 120, and 180, respectively. Each level of the cascade forest contains six Mondrian forests, and the number of decision trees in each Mondrian forest is randomly selected from 6, 12, 24, or 36.

According to the parameters of the Mondrian forest and the binary classification problem of predicting agricultural product web application users, during the multi-granularity scanning phase, each window outputs 1000, 500, and 200 probability feature vectors, resulting in a total of 1700-dimensional probability feature vectors fed into the cascade forest. In the cascade forest, the six Mondrian forests at the first level generate six-dimensional enhanced feature vectors, with cross-validation leaving five-dimensional feature vectors. These, combined with the 1700-dimensional probability feature vectors, form a 1705-dimensional transformed feature set for the input to the second level. This process continues until the model converges and the entire Mondrian forest training is completed.

We define TP as the number of actual purchasers predicted as purchasers, FP as the number of actual non-purchasers predicted as purchasers, FN as

the number of actual purchasers predicted as non-purchasers, and TN as the number of actual non-purchasers predicted as non-purchasers. Accuracy, precision, F1 score, and AUC (area under curve) are used to evaluate the performance of the agricultural behavior prediction.

Accuracy represents the probability of correctly predicting both purchasers and non-purchasers among the entire sample. The calculation expression is as follows:

$$C_r = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

Precision represents the probability that a predicted purchaser actually has purchasing behavior among those predicted as purchasers. The calculation expression is as follows:

$$R_r = \frac{TP}{TP + FP} \quad (11)$$

The F1 score combines both precision and recall to evaluate the behavior prediction method comprehensively. The calculation expression is as follows:

$$F_1 = \frac{R_r \cdot W_r}{R_r + W_r} \quad (12)$$

where W_r represents the recall of the prediction model, calculated as follows:

$$W_r = \frac{TP}{TP + FN} \quad (13)$$

The AUC (area under the receiver operating characteristic (ROC) curve) is a metric that ranges between 0 and 1. A higher AUC value indicates better predictive performance of the model.

5 Validity Verification

To validate the effectiveness of the Mondrian forest in predicting web application customer behavior, a simulation analysis was conducted using Python 3.5 and TensorFlow 2.2. The hardware specifications used were an Intel Core i7-7700U CPU with a clock speed of 2.2 GHz and 32 GB of RAM.

Logistic regression trees (LRT), extremely randomized trees (ERT), and Mondrian trees (MT) were used to construct DF models for predicting agricultural behavior. The accuracy of predictions under different decision trees is shown in Figure 2.

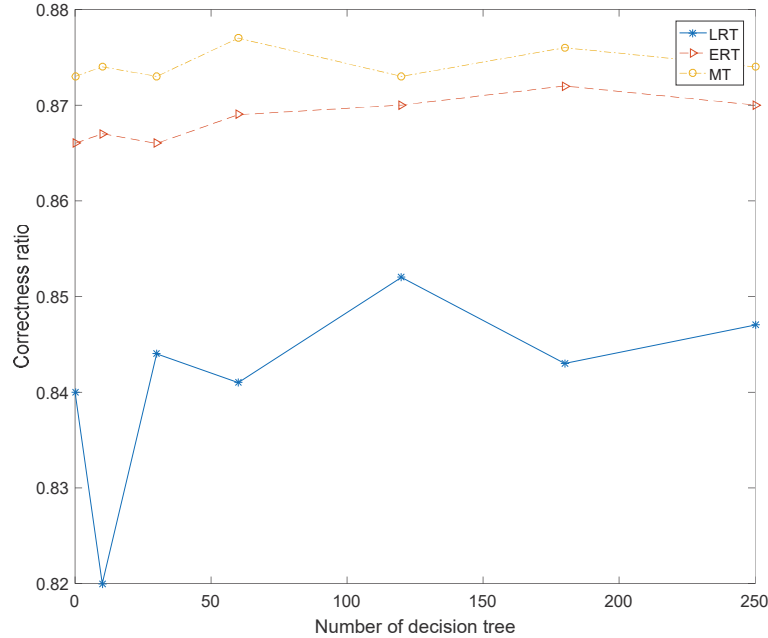


Figure 2 Accuracy of predictions with different decision trees.

As shown in Figure 3, with the increase in the number of decision trees in the random forest, the accuracy of the DF models constructed using different decision tree generation algorithms initially increases and then stabilizes in predicting agricultural behavior in the web application. Among them, the Mondrian tree (MT) decision tree generation model demonstrates superior accuracy in predicting agricultural purchasing behavior compared to the other two models.

Figure 3 illustrates the training time of the DF models with different decision trees. It is evident that the number of decision trees in the random forest is inversely proportional to the training efficiency of the DF model; as the number of decision trees increases, the training time of the model also increases. The experimental results indicate that the number of decision trees must balance the accuracy of DF predictions with training efficiency. Based on the experimental results, the number of decision trees for the LRT, ERT, and MT models was determined to be 10, 20, and 5, respectively.

To further assess the performance of the Mondrian DF in predicting agricultural behavior in the web application, a comparative experiment was

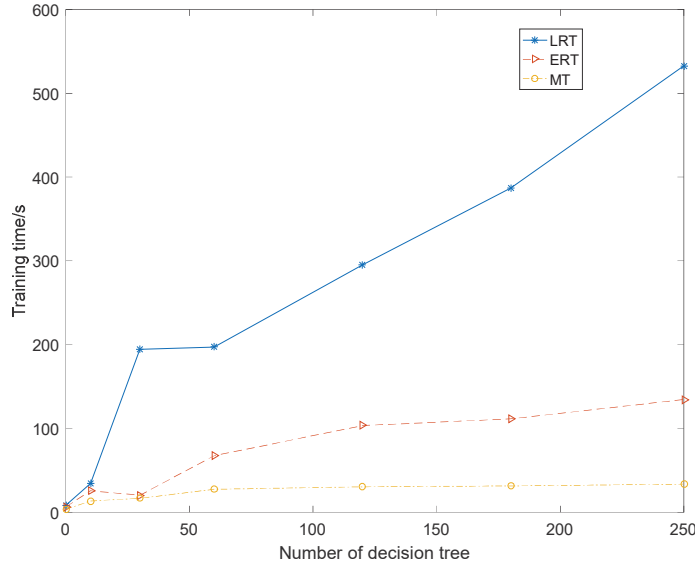


Figure 3 Training time of different decision tree models.

Table 1 Comparison of metrics for different prediction methods

Model	Accuracy (%)	Precision Represents (%)	F1 Value	AUC	Training Time (s)
SVM	87.32	91.65	0.832	0.743	184
XGB	91.24	92.11	0.920	0.802	140
RF	90.54	92.53	0.915	0.862	132
KNN	92.63	93.21	0.926	0.885	82
FNN	94.35	94.32	0.954	0.897	72
Mondrian deep forest	95.42	93.27	0.964	0.921	55

conducted with currently popular machine learning methods. The comparison methods include support vector machine (SVM), XGBoost decision trees (XGB), random forest (RF), K-nearest neighbors classification (KNN), and fuzzy neural network (FNN). The statistical results of various prediction methods for web application agricultural behavior prediction are shown in Table 1.

From Table 1, it can be seen that the Mondrian DF outperforms other prediction models in terms of accuracy, precision, F1 score, and AUC for predicting customer behavior in the agricultural product web application, while its training time is shorter than that of other models. This indicates

that the DF model can achieve higher prediction accuracy. A comprehensive comparison of the statistical results of various prediction models shows that the Mondrian DF model has a distinct advantage in predicting agricultural customer behavior in web applications. The experimental results validate the effectiveness of the proposed model.

6 Conclusions

This paper proposes the Mondrian DF model for predicting agricultural customer behavior in web applications. The Mondrian DF can achieve incremental learning by using Mondrian forests as the basic unit, improving predictive performance layer by layer through a cascade forest structure and adaptive mechanisms. The model was applied to predict customer behavior in an agricultural web application, and the dataset collected was used to train and test the prediction model. The comparison with support vector machine (SVM), XGBoost decision trees (XGB), random forest (RF), K-nearest neighbors (KNN), and fuzzy neural network (FNN) shows that the Mondrian DF model can overcome the difficulty of hyperparameter setting in traditional models and demonstrates superior overall performance. Since the Mondrian DF can cascade different models, such as replacing the cascade forest model with a linear regression model, there is potential for further improvement in classification prediction performance.

Acknowledgments

This research is supported by the “QingLan Project” of Jiangsu Province and Optimization of Delivery Algorithm based on Machine Vision (2023-ky38). We especially acknowledge Min Zhang for valuable discussions and the assistance with the experiments.

References

- [1] Silahtaroglu G, Donertasli H. Analysis and prediction of E-customers' behavior by mining clickstream data[C]. IEEE International Conference on Big Data. IEEE, 2015: 1466–1472.
- [2] Wen Wen, Lin Zetian, Cai Ruichu, et al. User Dynamics Preference Prediction Based on Embedding Learning[J]. Computer Science, 2019, 46(10):32–38.

- [3] Hu Xiaoli, Zhang Huibing, DONG Junchao, et al. Prediction of Repeated Purchase Behavior of New Users on E-commerce Platform Based on Ensemble Learning[J]. *Modern Electronics*, 2020, 43(11):115–119.
- [4] Strub F, Mary J. Collaborative filtering with stacked denoising autoencoders and sparse inputs[C]||NIPS workshop on machine learning for ecommerce, 2015.
- [5] Xu Z, Chen C, Lukasiewicz T, et al. Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling[C]||Proceedings of the 25th ACM International Conference on Information and Knowledge Management. ACM, 2016:1921–1924.
- [6] Wei J, He J, Chen K, et al. Collaborative filtering and deep learning based recommendation system for cold start items[J]. *Expert Systems with Applications*, 2017, 69:29–39.
- [7] Huang L W, Jiang B T, Lv S Y, et al. Survey on Deep Learning Based Recommender Systems[J]. *Chinese Journal of Computers*, 2018, 42(7):191–219.
- [8] Zeng X Y, Liu Q, Zhao H K, et al. Online Consumptions Prediction via Modeling User Behaviors and Choices[J]. *Journal of Computer Research and Development*, 2016, 53(8):1673–1683.
- [9] Ding Z, Qin Z, Zheng W T, et al. A Recommendation Model Based on Browsing Behaviors of Mobile Users [J]. *Journal of University of Electronic Science and Technology of China*, 2017, 46(6):907–912.
- [10] Timothy Dube, Mbulisi Sibanda, Onesimo Mutanga. Fine-scale characterization of irrigated and rainfed croplands at national scale using multi-source data, random forest, and deep learning algorithms, *ISPRS Journal of Photogrammetry and Remote Sensing*, 2023, 204:117–130.
- [11] Praveen Modi, Yugal Kumar. Smart detection and diagnosis of diabetic retinopathy using bat based feature selection algorithm and deep forest technique, *Computers & Industrial Engineering*, 2023, 182:109364.
- [12] Olsina, L., Garrido, A., Rossi, G., Distanto, D., and Canfora, G. Web Application Evaluation And Refactoring: a Quality oriented Improvement Approach. *Journal of Web Engineering*, 2008, 7(4): 258–280.
- [13] Nayak, A., Božić, B., and Longo, L. Data Quality Assessment and Recommendation of Feature Selection Algorithms: An Ontological Approach. *Journal of Web Engineering*, 2023, 22(01), 175–196.

- [14] Ashley Scillitoe, Pranay Seshadri, Mark Girolami, Uncertainty quantification for data-driven turbulence modelling with Mondrian forests, *Journal of Computational Physics*, 2021, 430:110116.
- [15] Wang Ying, Ruan Mengli. Simulation of risk assessment of abnormal transactions in e-commerce based on big data[J]. *Computer Simulation*, 2018, 35(3):369–372.
- [16] Li Xiaolei, Wang Yingtao, Xu Guowei, et al. Information Accurate Delivery System Based on User Access Trajectory[J]. *Information Technology*, 2020, 44(7):58–61.
- [17] Hu Xiaoli, Zhang Huibing, Dong Junchao, et al. Prediction of Repeated Purchase Behavior of New Users on E-commerce Platform Based on Ensemble Learning[J]. *Modern Electronics*, 2020, 43(11):115–119.
- [18] Li Zhengfang, Du Jinglin, Zhou Yun. Rainfall Prediction Model Based on Improved CART Algorithm[J]. *modern electronic technology*, 2020, 43(2):133–137.
- [19] Xue Canguan, Yan Xuefeng. Software Defect Prediction Based on Improved Deep Forest Algorithm[J]. *Computer Science*, 2018, 45(8):160–165.
- [20] Ren Jie, Hou Bojian, Jiang Yuan. Deep forest architecture based on multi-example learning[J]. *Computer Research and Development*, 2019, 56(8):1670–1676.

Biographies



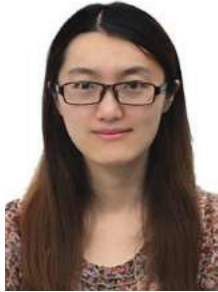
Chang-Sheng Ma received his Master's degree in Computer Application Technology from Soochow University, Suzhou, Jiangsu, P.R. China. He is currently an associate professor with the School of Information Engineering, Changzhou Vocational Institute of Mechatronic Technology. His research interests include Internet of Things engineering, computer vision, and deep learning.



Xiang-Ran Du received his M.Sc. degree from College of Mathematics and Computer Science, Hebei University. He works at Tianjin Maritime College. His main research interests include the application of the particle swarm optimization and neural network to the Chinese chess system or examination system and reinforcement learning to traffic control in urban areas.



Jing Lou received his Ph.D. degree in Computer Application Technology from Nanjing University of Science and Technology, Nanjing, Jiangsu, P.R. China. He is currently an associate professor with the School of Information Engineering, Changzhou Vocational Institute of Mechatronic Technology. His research interests include image processing, computer vision, and deep learning.



Ming-Qian Wang graduated from the School of Electronic Information, Jiangsu University of Science and Technology, where she received her M.Sc. degree in Control Theory and Control Engineering in April 2014. Since 2015, she has been teaching in Changzhou Electromechanical Vocational and Technical College, and currently serves as a lecturer, mainly engaged in the research of industrial Internet applications, network information security, and artificial intelligence. She has been selected as target audience for cultivating outstanding young backbone teacher in the “QingLan Project” of Jiangsu Province’s in 2023.