Research Article

An Application of a Three-Stage XGBoost-Based Model to Sales Forecasting of a Cross-Border E-Commerce Enterprise

Shouwen Ji,[1] Xiaojing Wang [ID],[1] Wenpeng Zhao,[2] and Dong Guo[3]

[1]*School of Traffic and Transportation, Beijing Jiaotong University, Haidian District, Beijing 100044, China*
[2]*Beijing Capital International Airport Co. Ltd., Beijing 100621, China*
[3]*School of Mechanical-Electronic and Vehicle Engineering, Beijing University of Civil Engineering and Architecture, Beijing 102600, China*

Correspondence should be addressed to Xiaojing Wang; 17120889@bjtu.edu.cn

Sales forecasting is even more vital for supply chain management in e-commerce with a huge amount of transaction data generated every minute. In order to enhance the logistics service experience of customers and optimize inventory management, e-commerce enterprises focus more on improving the accuracy of sales prediction with machine learning algorithms. In this study, a C-A-XGBoost forecasting model is proposed taking sales features of commodities and tendency of data series into account, based on the XGBoost model. A C-XGBoost model is first established to forecast for each cluster of the resulting clusters based on two-step clustering algorithm, incorporating sales features into the C-XGBoost model as influencing factors of forecasting. Secondly, an A-XGBoost model is used to forecast the tendency with the ARIMA model for the linear part and the XGBoost model for the nonlinear part. The final results are summed by assigning weights to forecasting results of the C-XGBoost and A-XGBoost models. By comparison with the ARIMA, XGBoost, C-XGBoost, and A-XGBoost models using data from Jollychic cross-border e-commerce platform, the C-A-XGBoost is proved to outperform than other four models.

1. Introduction

In order to enhance the logistics service experience of customers in the e-commerce industry chain, supply chain collaboration [1] requires that commodities are stocked in advance in local warehouses of various markets around the world, which can effectively reduce logistics time. However, for cross-border e-commerce enterprises, the production and sales areas of e-commerce products are globalized, which takes them longer to make preparations from the procurement of commodities, transportation, to customs quality inspection, etc. Therefore, algorithms and technologies of big data analysis are widely applied to predict sales of e-commerce commodities, which provide the data basis for the supply chain management and will provide key technical support for the global supply chain scheme of cross-border e-commerce enterprises.

Besides the large quantity and diversity of transaction data [2], sales forecasts are affected by many other factors due to the complexity of the cross-border e-commerce market [3, 4]. Therefore, to improve the precision and efficiency of forecasting, consideration of various factors in sales forecasting is still a challenge for e-commerce enterprises.

There are plenty of studies having been undertaken in sales forecasting. The methods of sales forecasts adopted in these studies can roughly be divided into time series models (TSMs) and machine learning algorithms (MLAs) [5, 6].

TSMs range from the exponential smoothing [7] to the ARIMA families [8], which have been used extensively to predict future trends by extrapolating based on historical observation data. Although TSMs have been proven to be useful for sales forecasting, their forecasting ability is limited by their assumption of a linear behavior [9], and they do not take external factors such as price changes and promotions

into account [10]. Therefore, univariate forecasting methods are usually adopted as a benchmark model in many studies [11, 12].

Another important branch of forecasting has been MLAs. The existing MLAs have been largely influenced by state-of-the-art forecasting techniques, which range from artificial neural network (ANN), convolutional neural network (CNN), radial basis function (RBF), long short-term memory network (LSTM), extreme learning machine (ELM) to support vector regression (SVR), etc. [13].

On the one hand, some existing forecasting models have made comparisons between MLAs and TSMs [14]. Ansuj et al. showed the superiority of ANN on the ARIMA method in sales forecasting [15]. Alon et al. compared ANN with traditional methods, including Winters exponential smoothing, Box–Jenkins ARIMA model, and multivariate regression, indicating that ANNs perform favorably in relation to the more traditional statistical methods [16]. Di Pillo et al. assessed the application of SVM to sales forecasting under promotion impacts, which was compared with ARIMA, Holt-Winters, and exponential smoothing [17].

On the other hand, MLAs based on TSMs have also been applied in sales prediction. Wang et al. proved the advantages of the integrated model combining ARIMA with ANN in modeling the linear and nonlinear parts of the data set [18]. In [19], an ARIMA forecasting model was established and the residual of the ARIMA model was trained and fitted by the BP neural network. A novel LSTM ensemble forecasting algorithm was presented by Choi and Lee [20] that effectively combines multiple forecast results from a set of individual LSTM networks. In order to better handle irregular sales patterns and take various factors into account, some algorithms have been attempted to exploit more information in sales forecasting as an increasing amount of data are becoming available in e-commerce. Zhao and Wang [21] provided a novel approach to learning effective features automatically from structured data using CNN. Bandara et al. attempted to incorporate sales demand patterns and cross-series information in a unified model by training the LSTM model [22]. More importantly, ELM was widely applied in forecasting. Luo et al. [23] proposed a novel data-driven method to predict user behavior by using ELM with distribution optimization. In [24], ELM was enhanced under deep learning framework to forecast wind speed.

Although there are various methods of forecasting, the choice of methods is determined by the characteristics of different goods [25]. Kulkarni et al. [26] argued that product characteristics could have an impact on both searching and sales due to the characteristics inherent to products were the main attributes that potential consumers were interested in. Therefore, to better reflect the characteristics of goods into sales forecasting, clustering techniques have been introduced to forecast [27]. For example, in [28, 29], both fuzzy neural networks and clustering methods were used to improve the results of neural networks. Lu and Wang [30] constructed the SVR to deal with the demand forecasting problem with the aid of the hierarchical self-organizing maps and independent component analysis. Lu and Kao [31] put forward a sales forecasting method based on clustering using extreme

learning machine and combination linkage method. Dai et al. [32] built a clustering-based sales forecasting scheme based on SVR. A clustering-based forecasting model by combining clustering and machine learning methods was developed by Chen and Lu [33] for computer retailing sales forecasting.

According to the above literature review, a three-stage XGBoost-based forecasting model is constructed to focus on the two aspects (the sales features and tendency of a data series) mentioned above in this study.

Firstly, in order to forecast the sales features, various influencing factors of sales are first introduced in this study by the two-step clustering algorithm [34], which is an improved algorithm based on BIRCH [35]. Then, a C-XGBoost model based on clustering is presented to model for each cluster of the resulting clusters with the XGBoost algorithm, which has been proved to be an efficient predictor in many data analysis contests such as Kaggle and in many recent studies [36, 37].

Secondly, to achieve higher predicting accuracy in the tendency of data series, an A-XGBoost model is presented integrating the strengths of the ARIMA and XGBoost model, respectively, for the linear part and the nonlinear part of data series. Therefore, a C-A-XGBoost model is constructed as the final combination model by weighting for the C-XGBoost and A-XGBoost models, which takes the multiple factors affecting the sales of goods and the trend of the time series into account.

The paper is organized into 5 sections, the rest of which is organized as follows: In Section 2, the key models and algorithms employed in the study are shortly described, including the feature selection, two-step clustering algorithm, a method of parameter determination of the ARIMA, and the XGBoost. In Section 3, a three-stage XGBoost-based model is proposed to forecast both the sales features and tendency of time series. In Section 4, numerical examples are used to illustrate the validity of the proposed forecasting model. In Section 5, the conclusions along with a note regarding future research directions are summarized.

2. Methodologies

2.1. Feature Selection. With the emergence of web technologies, there is an ever-increasing growth in the amount of big data in the e-commerce environment [38]. Variety is one of the critical attributes in big data as they are generated from a wide variety of sources and formats, including text, web, tweet, audio, video, click-stream, and log files [39]. In order to remove most irrelevant and redundant information from various data, many techniques of feature selection (removing variables that are irrelevant) and feature extraction (applying some transformations to the existing variables to obtain a new one) have been discussed to reduce the dimensionality of the data [40], including filter-based and wrapper feature selection. Wrapper feature selection employs a subroutine statistical resampling technique (such as cross-validation) in the actual learning algorithm to forecast the accuracy of feature subsets [41], which is a better choice for different algorithms modeling the different data

series. Instead, filter-based feature selection is suitable for different algorithms, modeling the same data series [42].

In this study, wrapper feature selection in the forecasting and clustering algorithms is directly applied to removing unimportant attributes in multidimensional data based on standard deviation (SD), the coefficient of variation (CV), Pearson correlation coefficient (PCC), and feature importance scores (FIS), of which the details are as follows.

SD reflects the degree of dispersion of data set, which is calculated as σ, where N and μ denote the number of samples and mean value of the sample x_i, respectively:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2}. \tag{1}$$

CV is a statistic to measure the degree of variation of observed values in the data which is calculated as c_v:

$$c_v = \frac{\sigma}{\mu}. \tag{2}$$

PCC is a statistic used to reflect the degree of linear correlation between two variables, which is calculated as r:

$$r = \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{X_i - \overline{X}}{\sigma_X} \right) \left(\frac{Y_i - \overline{Y}}{\sigma_Y} \right), \tag{3}$$

where $((X_i - \overline{X})/\sigma_X)$, $\overline{X}$, and σ_X represent the standard deviation, mean value, and standard score of X_i.

FIS provides a score indicating how useful or valuable each feature is in the construction of the boosted decision trees within the model. The more an attribute is used to make key decisions with decision trees, the higher its relative importance [43]. The importance is calculated for a single decision tree by the performance measure increased by each attribute split point, weighted by the number of observations the node is responsible for. The performance measure may be the purity such as the Gini Index [44] used to select the split points or another more specific error function. The feature importance is then averaged across all of the decision trees within the model [45].

2.2. Two-Step Clustering Algorithm.

Clustering aims at partitioning samples into several disjoint subsets, making samples in the same subsets highly similar to each other [46]. The most widely applied clustering algorithms can broadly be categorized as the partition, hierarchical, density-based, grid-based, and model-based methods [47, 48].

The selection of clustering algorithms mainly depends on the scale and the type of collected data. Clustering can be conducted using traditional algorithms when dealing with numeric or categorical data [49, 50]. The BIRCH, as one of the hierarchical methods, introduced by Zhang et al. [35] is especially suitable for the large data sets of continuous attributes [51]. But in case of the large and mixed data, the two-step clustering algorithm in SPSS Modeler is advised in this study. The two-step clustering algorithm is a modified method based on BIRCH setting the log-likelihood distance as the measure, which can measure the distance between

continuous data and the distance between categorical data [34]. Similar to BIRCH, the two-step clustering algorithm first performs a preclustering step of scanning the entire data set and storing the dense regions of data records in terms of summary statistics. A hierarchical clustering algorithm is then applied to clustering the dense regions. Apart from the ability to handle the mixed type of attributes, the two-step clustering algorithm differs from BIRCH in automatically determining the appropriate number of clusters and a new strategy of assigning cluster membership to noisy data.

As one of the hierarchical algorithms, the two-step clustering algorithm is also more efficient in handling noise and outliers than partition algorithms. More importantly, it has unique advantages over other algorithms in the automatic mechanism of determining the optimal number of clusters. Therefore, with regard to large and mixed transaction data sets of e-commerce, two-step clustering algorithm is a reliable choice for clustering goods, of which the key technologies and processes are illustrated in Figure 1.

2.2.1. Preclustering.
The clustering feature (CF) tree growth in the BIRCH algorithm is used to read data records in data set one by one, in the process of which the handling of outliers is implemented. Then, subclusters C_j are obtained from data records in dense areas while generating a CF tree.

2.2.2. Clustering.
Take the subclusters C_j as the object, the clusters C_J are obtained by merging the subclusters one by one based on agglomerative hierarchical clustering methods [52] until the optimal number of clusters is determined based on the minimum value of Bayesian information criterion (BIC).

2.2.3. Cluster Membership Assignment.
The data records are assigned to the nearest clusters by calculating the log-likelihood distance between the data records and subclusters of the clusters C_J.

2.2.4. Validation of the Results.
The performance of clustering results is measured by silhouette coefficient S, where a is the mean distance between the sample and its cluster and b is the mean distance between the sample and its different cluster. The higher the value of S is, the better the clustering result is:

$$S = \frac{b - a}{\max(a, b)}. \tag{4}$$

2.3. Parameter Determination of ARIMA Model.
ARIMA models obtained from a combination of autoregressive and moving average models [53]. The Box–Jenkins methodology in time series theory is applied to establish an ARIMA (p, d, q) model, and its calculation steps can be found in [54]. The ARIMA has limitations in determining parameters because its parameters are usually determined based on plots of ACF and PACF, which usually leads to the judging deviation. However, a function named auto.arima ( ) in R package
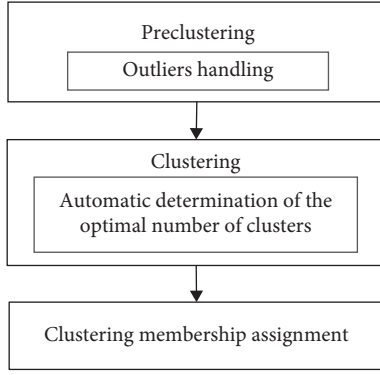
FIGURE 1: The key technologies and processes of the two-step clustering algorithm.

"forecast" [55] is used to automatically generate an optimal ARIMA model for each of the time series based on the smallest Akaike information criterion (AIC) and BIC [56], which makes up for the disadvantage of ARIMA during judging parameters.

Therefore, a combined method of parameter determination is proposed to improve the fitting performance of the ARIMA, which combines the results of ACF and PACF plots with that of the auto.arima ( ) function. The procedures are illustrated in Figure 2 and described as follows:

> *Step 1.* Test the stationary and white noise by the augmented Dickey–Fuller (ADF) and Box–Pierce tests before modeling ARIMA. If both stationarity and white noise tests are passed, the ARIMA is suitable for the time series.
>
> *Step 2.* Determine a part of parameter combinations based on ACF and PACF plots, and determine another part of parameter combinations by the auto.arima ( ) function in R application.
>
> *Step 3.* Model the ARIMA under different parameter combinations, and then calculate the values of AIC for different models.
>
> *Step 4.* Determine the optimal parameters combination of the ARIMA with the minimum of AIC.

2.4. XGBoost Algorithm.
The XGBoost is short for "Extreme Gradient Boosting" proposed by Friedman [57]. As the relevant basic theory of the XGBoost has been mentioned in plenty of previous papers [58, 59], the procedures of the algorithm [60] are covered in this study rather than basic theory.

2.4.1. Feature Selection.
The specific steps of feature selection via the XGBoost are as follows: data cleaning, data feature extraction, and data feature selection based on the scores of feature importance.

2.4.2. Modeling Training.
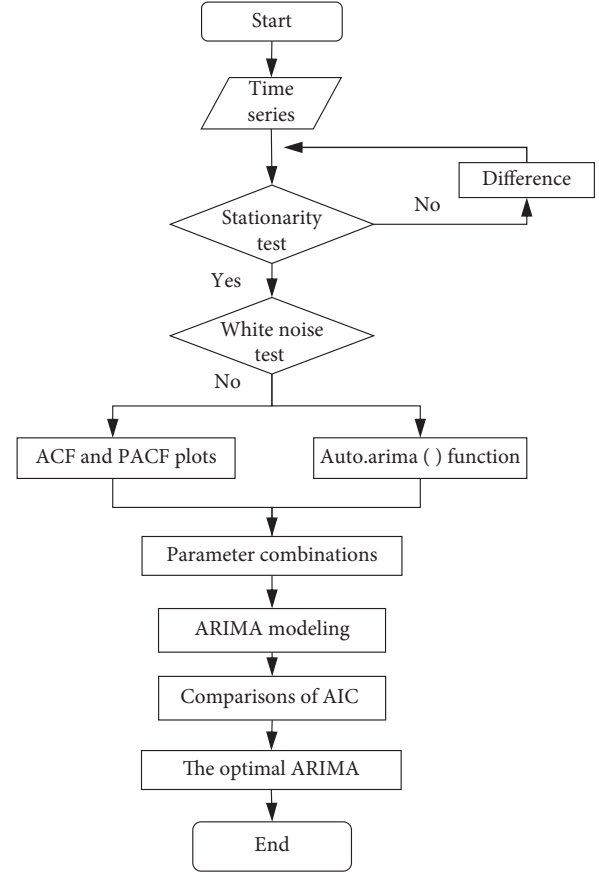The model is trained based on the selected features with default parameters.

FIGURE 2: The procedures for parameter determination of the ARIMA model.

2.4.3. Parameter Optimization.
Parameter optimization is aimed at minimizing the errors between predicted values and actual values. There are three types of parameters in the algorithm, of which the descriptions are listed in Table 1.

The general steps of determining the hyperparameter of the XGBoost model are as follows:

> *Step 1.* The number of estimators is firstly tuned to optimize the XGBoost when fixing the learning rate and other parameters
>
> *Step 2.* Different combinations of max_depth and min_child_weight are tuned to optimize the XGBoost
>
> *Step 3.* Max delta step and Gamma is tuned to make the model more conservative with the determined parameter in Step 1 and Step 2
>
> *Step 4.* Different combinations of subsample and colsample_bytree are tuned to prevent overfitting
>
> *Step 5.* Regularization parameters are increased to make the model more conservative
>
> *Step 6.* The learning rate is reduced to prevent overfitting

3. The Proposed Three-Stage Forecasting Model

In this research, a three-stage XGBoost-based forecasting model, named C-A-XGBoost model, is proposed in

TABLE 1: The description of parameters in the XGBoost model.

Type of parameters	Parameters	Description of parameters	Main purpose
Booster parameters	Max depth	Maximum depth of a tree	Increasing this value will make the model more complex and more likely to be overfit
	Min_child_weight	Minimum sum of weights in a child	The larger the min_child_weight is, the more conservative the algorithm will be
	Max delta step	Maximum delta step	It can help make the update step more conservative
	Gamma	Minimum loss reduction	The larger the gamma is, the more conservative the algorithm will be
	Subsample	Subsample ratio of the training instances	It is used in the update to prevent overfitting
	Col sample by a tree	Subsample ratio of columns for each tree	It is used in the update to prevent overfitting
	Eta	Learning rate	Step size shrinkage used in the update can prevent overfitting
Regularization parameters	Alpha Lambda	Regularization term on weights	Increasing this value will make the model more conservative
Learning task parameters	Reg: linear	Learning objective	It is used to specify the learning task and the learning objective
Command line parameters	Number of estimators	Number of estimators	It is used to specify the number of iterative calculations

consideration of both the sales features and tendency of data series.

In Stage 1, a novel C-XGBoost model is put forward based on the clustering and XGBoost, which incorporates different clustering features into forecasting as influencing factors. The two-step clustering algorithm is first applied to partitioning commodities into different clusters based on features, and then each cluster in the resulting clusters is modeled via XGBoost.

In Stage 2, an A-XGBoost model is presented by combining the ARIMA with XGBoost to predict the tendency of time series, which takes the strength of linear fitting ability of ARIMA and the strong nonlinear mapping ability of XGBoost. ARIMA is used to predict the linear part, and the rolling prediction method is employed to establish XGBoost to revise the nonlinear part of the data series, namely, residuals of the ARIMA.

In Stage 3, a combination model is constructed based on C-XGBoost and A-XGBoost, named C-A-XGBoost. The C-A-XGBoost is aimed at minimizing the sum errors of squares by assigning weights to the results of C-XGBoost and A-XGBoost, in which the weights reflect the reliability and credibility of sales features and tendency of data series.

The procedures of the proposed three-stage model are demonstrated in Figure 3, of which the details are given as follows.

3.1. Stage 1. C-XGBoost Model.

The two-step clustering algorithm is applied to clustering a data series into several disjoint clusters. Then, each cluster in the resulting clusters is set as the input and output sets to construct and optimize the corresponding C-XGBoost model. Finally, testing samples are partitioned into the corresponding cluster by the trained two-step clustering model, and then the prediction results

are calculated based on the corresponding trained C-XGBoost model.

3.2. Stage 2. A-XGBoost Model.

The optimal ARIMA based on the minimum of AIC after the data series pass the tests of stationarity and white noise is trained and determined, of which the processes are described in Section 2. Then, the residual vector $\mathbf{e} = (r_1, r_2, \ldots, r_n)^{\mathrm{T}}$ between the predicted values and actual values are obtained by the trained ARIMA model. Next, the A-XGBoost is established by setting columns from 1 to k, and column $(k+1)$ in $\mathbf{R}$ as the input and output, respectively, as is illustrated in the following equation:

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 & \cdots & r_k & r_{k+1} \\ r_2 & r_3 & \cdots & r_{k+1} & r_{k+2} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ r_{n-k-1} & r_{n-k} & \cdots & r_{n-2} & r_{n-1} \\ r_{n-k} & r_{n-k+1} & \cdots & r_{n-1} & r_n \end{bmatrix}_{(n-k)\times(k+1)}. \quad (5)$$

The final results of the test set are calculated by summing the predicted results of the linear part by the trained ARIMA and that of residuals with the established XGBoost.

3.3. Stage 3. C-A-XGBoost Model.

In this stage, a combination strategy is explored to minimize the error sum of squares MSE in equation (6) by assigning weights w_C and w_A to C-XGBoost and A-XGBoost, respectively. The predicted results are calculated using equation (7), where $\widehat{Y}_{CA}(k)$, $\widehat{y}_C(k)$, and $\widehat{y}_A(k)$ denote the corresponding forecast values of the k-th sample via C-XGBoost, A-XGBoost, and C-A-XGBoost, respectively. In equation (6), $y(k)$ is the actual value of the k-th sample:
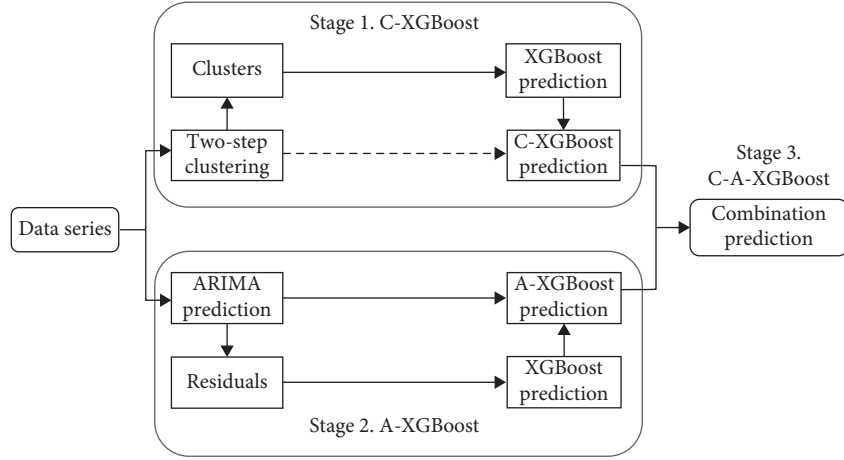
FIGURE 3: The procedures of the proposed three-stage model.

$$\min \text{ MSE} = \frac{1}{n}\sum_{k=1}^{n}\left[\hat{Y}_{CA}(k) - y(k)\right]^2, \quad (6)$$

$$\widehat{Y}_{CA}(k) = w_C\widehat{y}_C(k) + w_A\widehat{y}_A(k). \quad (7)$$

The least squares are employed in exploring the optimal weights (w_C and w_A), the calculation of which is simplified by transforming the equations into the following matrix operations.

In equation (8), the matrix **B** consists of the predicted values of C-XGBoost and A-XGBoost.

In equation (9), the matrix **W** consists of the weights.

In equation (10), the matrix **Y** consists of the actual values.

Equation (11) is obtained by transforming the equation (7) into the matrix form.

Equation (12) is calculated based on equation (11) left multiplying by the transpose of the matrix **B**.

According to equation (13), the optional weights (w_C and w_A) are calculated.

$$\mathbf{B} = \begin{bmatrix} \hat{y}_C(1) & \hat{y}_A(1) \\ \hat{y}_C(2) & \hat{y}_A(2) \\ \vdots & \vdots \\ \hat{y}_C(n) & \hat{y}_A(n) \end{bmatrix}, \quad (8)$$

$$\mathbf{W} = \begin{bmatrix} w_C \\ w_A \end{bmatrix}, \quad (9)$$

$$\mathbf{Y} = [y(1), y(2), \ldots, y(n)], \quad (10)$$

$$\mathbf{BW} = \mathbf{Y}, \quad (11)$$

$$\mathbf{B^T BW} = \mathbf{B^T Y}, \quad (12)$$

$$\mathbf{W} = \left(\mathbf{B^T B}\right)^{-1}\mathbf{B^T Y}. \quad (13)$$

4. Numerical Experiments and Comparisons

4.1. Data Description.
To illustrate the effectiveness of the developed C-A-XGBoost model, the following data series are used to verify the forecasting performance.

4.1.1. Source Data Series.
As listed in Table 2, there are eight data series in source data series. The data series range from Mar. 1, 2017 to Mar. 16, 2018.

4.1.2. Clustering Series.
There are 10 continuous attributes and 6 categorical attributes in clustering series, which are obtained by reconstructing the source data series. The attribute descriptions of the clustering series are illustrated in Table 3.

4.2. Uniform Experimental Conditions.
To verify the performance of the proposed model according to performance evaluation indexes, some uniform experimental conditions are established as follows.

4.2.1. Uniform Data Set.
As shown in Table 4, the data series are partitioned into the training set, validation set, and test set so as to satisfy the requirements of different models. The data application is described as follows:

(1) The clustering series cover samples of 381 days.

(2) For the C-XGBoost model, training set 1, namely, samples of the first 347 days in clustering series, is utilized to establish the two-step clustering models. The resulting samples of two-step clustering are used to construct XGBoost models. The test set with the remaining samples of 34 days is selected to validate the C-XGBoost model. In detail, the test set is first partitioned into the corresponding clusters by the established two-step clustering model, and then the test set is applied to checking the validity of the corresponding C-XGBoost models.

TABLE 2: The description of source data series.

Data series	Fields
Customer behavior data[a]	Data date; goods click; cart click; favorites click
Goods information data[b]	Goods id; SKU[i] id; level; season; brand id
Goods sales data[c]	Data date; SKU sales; goods price; original shop price
The relationship between goods id and SKU id[d]	Goods id; SKU id
Goods promote price[e]	Data date; goods price; goods promotion price
Marketing[f]	Data date; marketing; plan
Holidays[g]	Data date; holiday
Temperature[h]	Data date; temperature mean

[a–f]The six data series are sourced from the historical data of the Saudi Arabian market in Jollychic cross-border e-commerce trading platform (https://www.jollychic.com/). [g]The data of holidays are captured from the URL http://shijian.cc/114/jieri2017/. [h]The data of temperature are captured from the URL https://www.wunderground.com/weather/eg/saudi-arabia. [i]SKU's full name is stock keeping unit. Each product has a unique SKU number.

TABLE 3: The description of clustering series.

Fields	Meaning of fields	Fields	Meaning of fields
Data date	Date	Favorites click	Number of clicks on favorites
Goods code	Goods code	Sales unique visitor	Number of unique visitors
SKU code	SKU code	Goods season	Seasonal attributes of goods
SKU sales	Sales of SKU	Marketing	Activity type code
Goods price	Selling price	Plan	Activity rhythm code
Original shop price	Tag price	Promotion	Promotion code
Goods click	Number of clicks on goods	Holiday	The holiday of the day
Cart click	Number of clicks on purchasing carts	Temperature mean	Mean of air temperatures (°F)

(3) For A-XGBoost model, the training set 2 with the samples of 1st–277th days are used to construct the ARIMA, and the validation set is used to calculate the residuals of ARIMA forecast, which are used to train the A-XGBoost model. Then, the test set is employed to verify the performance of the model.

(4) The test set had the final 34 data samples, which are employed to fit the optimal combination weights for C-XGBoost and A-XGBoost models.

4.2.2. Uniform Evaluation Indexes. Several performance measures have previously been applied to verifying the viability and effectiveness of forecasting models. As illustrated in Table 5, the common evaluation measurements are chosen to distinguish the optimal forecasting model. The smaller they are, the more accurate the model is.

4.2.3. Uniform Parameters of the XGBoost Model. The first priority for optimization is to tune depth and min_child_weight with other parameters fixed, which are the most effective way for optimizing the XGBoost. The ranges of depth and child weigh are 6–10 and 1–6, respectively. Default values of parameters are listed in Table 6.

4.3. Experiments of C-A-XGBoost Model

4.3.1. C-XGBoost Model

(1) Step 1. Commodity clustering: The two-step clustering algorithm is first applied to training set 1. Standardization applies to the continuous attributes; the noise percent of outliers handling is 25%; log-likelihood distance is the basis of distance measurement; BIC is set as the clustering criterion.

As shown in Figure 4, the clustering series are partitioned into 12 homogeneous clusters based on 11 features, denoted as $C12\_j$, $(j = 1, 2, \ldots, 12)$, and the silhouette coefficient is 0.4.

As illustrated in Figure 5, the ratio of sizes is 2.64 and the percentage is not too large or too small for each cluster. Therefore, cluster quality is acceptable.

(2) Step 2. Construct the C-XGBoost models: Features are first selected from each cluster $C12\_j$ of the 12 clusters based on feature importance scores. After that, setting the selected features of each cluster and SKU sales in Table 3 as the input and output varieties, respectively, the C-XGBoost models are constructed for each cluster $C12\_j$, denoted as $C12\_j\_$XGBoost.

Take the cluster $C12\_3$ in the 12 clusters as an example to illustrate the processes of modeling XGBoost.

For $C12\_3$, the features listed in Table 3 are first filtered and the 7 selected features are displayed in Figure 6. It can be observed that F1 (goods click), F3 (cart click), F5 (goods price), F6 (sales unique visitor), and F7 (original shop price) are the dominating factors. However, F2 (temperature mean) and F4 (favorites click) have fewer contributions to the prediction.

Setting the 11 features of the cluster $C12\_3$ in Step 1 and the corresponding SKU sales in Table 3 as the input and output, respectively, the $C12\_3\_$XGBoost is pretrained under the default parameters in Table 6. For the prebuilt $C12\_3\_$XGBoost model, the value of ME is 0.393 and the value of MAE is 0.896.

(3) Step 3. Parameter optimization: XGBoost is an algorithm with supervised learning, so the key to optimization is to

TABLE 4: The description of the training set, validation set, and test set.

Data set	Samples	Number of weeks	Start date	End date	The first day	The last day
Training set 1	Clustering series	50	Mar.1, 2017 (WED)	Dec.2, 2017 (SAT)	1	347
Training set 2	SKU code = 94033	50	Mar.1, 2017 (WED)	Dec.2, 2017 (SAT)	1	277
Validation set	SKU code = 94033	10	Dec.3, 2017 (SUN)	Feb.10, 2018 (SAT)	278	347
Test set	SKU code = 94033	5	Feb.11, 2018 (SUN)	Mar.16, 2018 (FRI)	348	381

TABLE 5: The description of evaluation indexes.

Evaluation indexes	Expression	Description
ME	$\mathrm{ME} = 1/(b - a + 1)\sum_{k=a}^{b}(y_k - \hat{y}_k)$	The mean sum error
MSE	$\mathrm{MSE} = 1/(b - a + 1)\sum_{k=a}^{b}(y_k - \hat{y}_k)^2$	The mean squared error
RMSE	$\mathrm{RMSE} = \sqrt{1/(b - a + 1)\sum_{k=a}^{b}(y_k - \hat{y}_k)^2}$	The root mean squared error
MAE	$\mathrm{MAE} = 1/(b - a + 1)\sum_{k=a}^{b}|y_k - \hat{y}_k|$	The mean absolute error

y_k is the sales of the k-th sample. $\hat{y}_k$ denotes the corresponding prediction.

TABLE 6: Default parameters values of XGBoost.

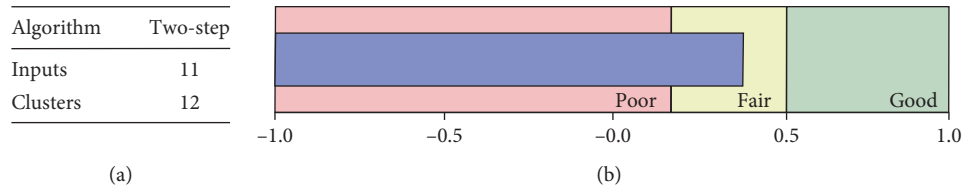| Parameters	Number of estimators	Max depth	Min_child_weight	Max delta step	Objective	Subsample	Eta
Default value	100	6	1	0	Reg: linear	1	0.3
Parameters	Gamma	Col sample by tree	Col sample by level	Alpha	Lambda	Scale position weight	
Default value	0.1	1	1	0	1	1	

FIGURE 4: Model summary and cluster quality of the two-step clustering model. (a) The summary of the two-step clustering model. (b) The Silhouette coefficient of cohesion and separation for 12 clusters.
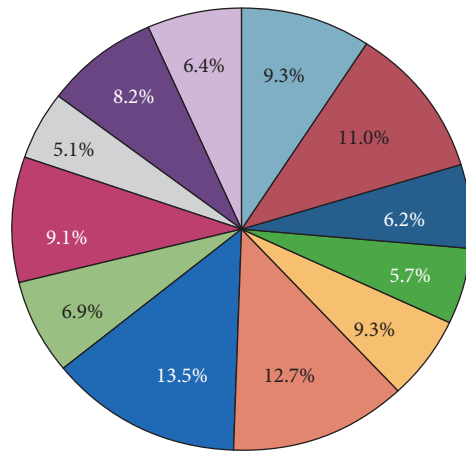
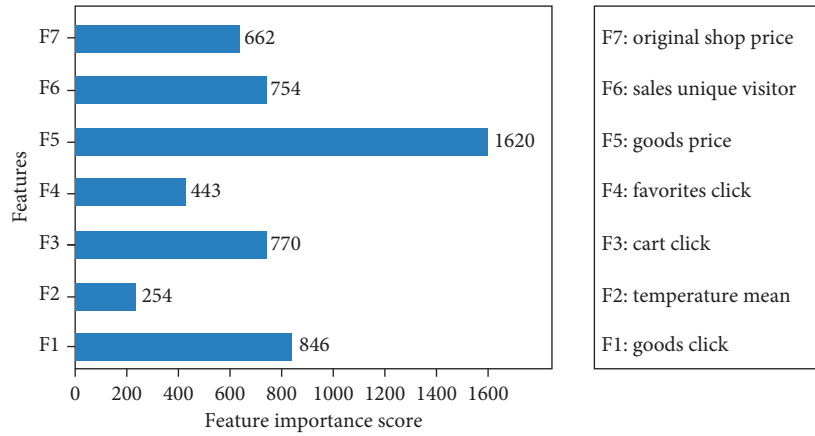FIGURE 5: Cluster sizes of the clustering series by two-step clustering algorithm.

FIGURE 6: Feature importance score of the C12_3_XGBoost model.

determine the appropriate input and output variables. In contrast, parameter optimization has less impact on the accuracy of the algorithm. Therefore, in this paper, only the primary parameters including max_depth and min_child_weight are tuned to optimize the XGBoost [61]. The model can achieve a balanced point because increasing the value of max_depth will make the model more complex and more likely to be overfit, but increasing the value of min_child_weight will make the model more conservative.

The prebuilt C12_3_XGBoost model is optimized to minimize ME and MAE by tuning max_depth (from 6 to 10) and min_child_weight (from 1 to 6) when other parameters are fixed, in which the ranges of parameters are determined according to lots of case studies with the XGBoost such as [62]. The optimal parameter combination is determined by the minimum of the ME and MAE under different parameter combination.

Figure 7 shows the changes of ME and MAE based on XGBoost as depths and min_child_weight change. It can be seen that both the ME and MAE are the smallest when depth is 9 and min_child_weight is 2. That is, the model is optimal.

(4) Step 4. Results on the test set: The test set is partitioned into the corresponding clusters by the trained two-step clustering model in Step 1. After that, the Steps 2-3 are repeated for the test set.

As shown in Table 7, the test set is partitioned into the clusters C12_3 and C12_4. Then, the corresponding models C12_3_XGBoost and C12_4_XGBoost are determined. C12_3_XGBoost has been trained and optimized as an example in Steps 2-3, and the C12_4_XGBoost is also trained and optimized by repeating Steps 2-3. Finally, the prediction results are obtained by the optimized C12_3_XGBoost and C12_4_XGBoost.

As illustrated in Figure 8, ME and MAE for C12_4_XGBoost change with the different values of depth and min_child_weight. The model performs the best when depth is 10 and min_child_weight is 2 because both the ME and MAE are the smallest. The forecasting results of the test set are calculated and summarized in Table 7.

4.3.2. A-XGBoost Model

(1) Step 1. Test stationarity and white noise of training set 2: For training set 2, the p value of the ADF test and Box–Pierce test are 0.01 and 3.331×10^{-16}, respectively, which are lower than 0.05. Therefore, the time series is stationary and nonwhite noise, indicating that training set 2 is suitable for the ARIMA.

(2) Step 2. Train ARIMA model: According to Section 2.3, parameter combinations are firstly determined by ACF and PACF plots, and auto.arima ( ) function in R package "forecast."

As shown in Figure 9(a), SKU sales have a significant fluctuation in the first 50 days compared with the sales after 50 days; in Figure 9(b), the plot of ACF has a high trailing characteristic; in Figure 9(c), the plot of PACF has a decreasing and oscillating phenomenon. Therefore, the first-order difference should be calculated.

As illustrated in Figure 10(a), SKU sales fluctuate around zero after the first-order difference. Figures 10(b) and 10(c) graphically present plots of ACF and PACF after the first-order difference, both of which have a decreasing and oscillating phenomenon. It indicates that the training set 2 conforms to the ARMA.

As a result, the possible optimal models are ARIMA (2, 1, 2), ARIMA (2, 1, 3), and ARIMA (2, 1, 4) according to the plots of ACF and PACF in Figure 10.

Table 8 shows the AIC values of the ARIMA under different parameters, which are generated by the auto.arima ( ) function. It can be concluded that the ARIMA (0, 1, 1) is the best model because its AIC has the best performance.

To further determine the optimal model, the AIC and RMSE of ARIMA models under different parameters are summarized in Table 9. The possible optimal models include the 3 possible optimal ARIMA judged by Figure 10 and the best ARIMA generated by the auto.arima ( ) function. According to the minimum principles, the ARIMA (2, 1, 4) is optimal because both AIC and RMSE have the best performance.

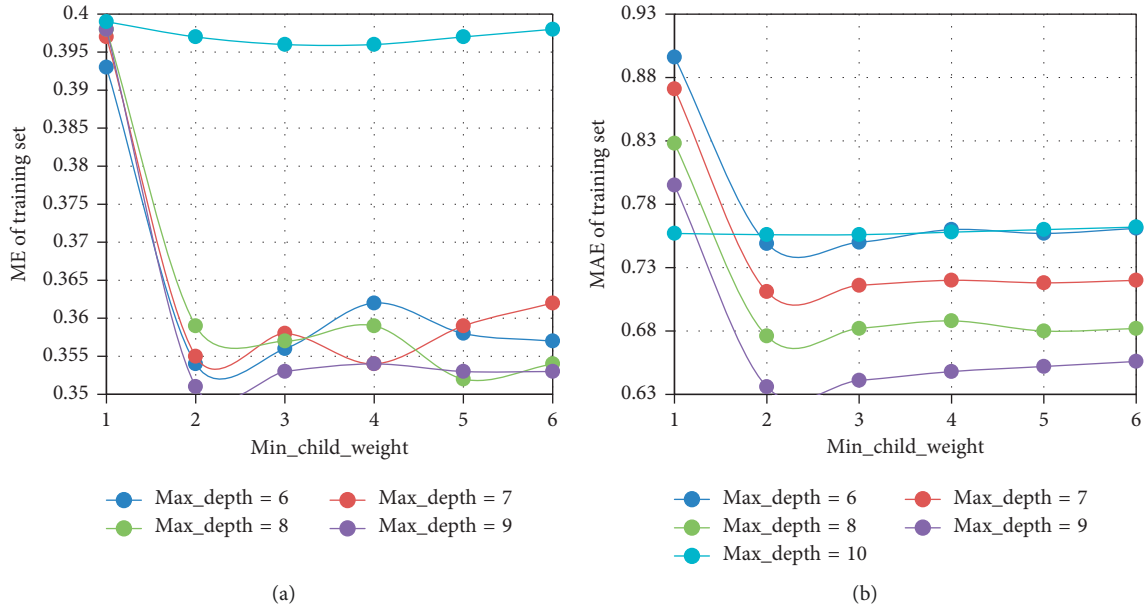(a)   (b)

FIGURE 7: ME and MAE of C12_3_XGBoost under different parameters. (a) Mean error of the training set. (b) Mean absolute error of the training set.

TABLE 7: The results of C-XGBoost for the test set.

Test set	Days	C12_j	C12_j_XGBoost model	Depth and min_child_weight	Training set 1		Test set	
					ME	MAE	ME	MAE
348th–372th	25	3	C12_3_XGBoost model	(9, 2)	0.351	0.636	4.385	4.400
373rd–381st	9	4	C12_4_XGBoost model	(10, 2)	0.339	0.591	1.778	2.000
348th–381st	34	—	—	—	—	—	3.647	3.765

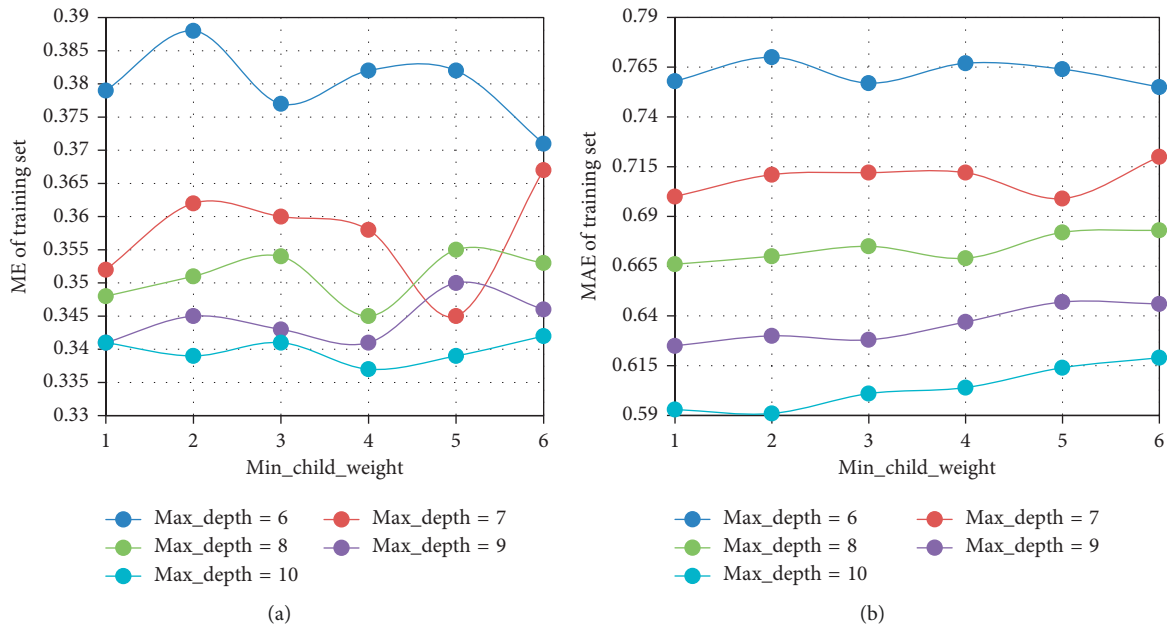(a)   (b)

FIGURE 8: ME and MAE of C12_4_XGBoost under different parameters. (a) Mean error of the training set. (b) Mean absolute error of the training set.
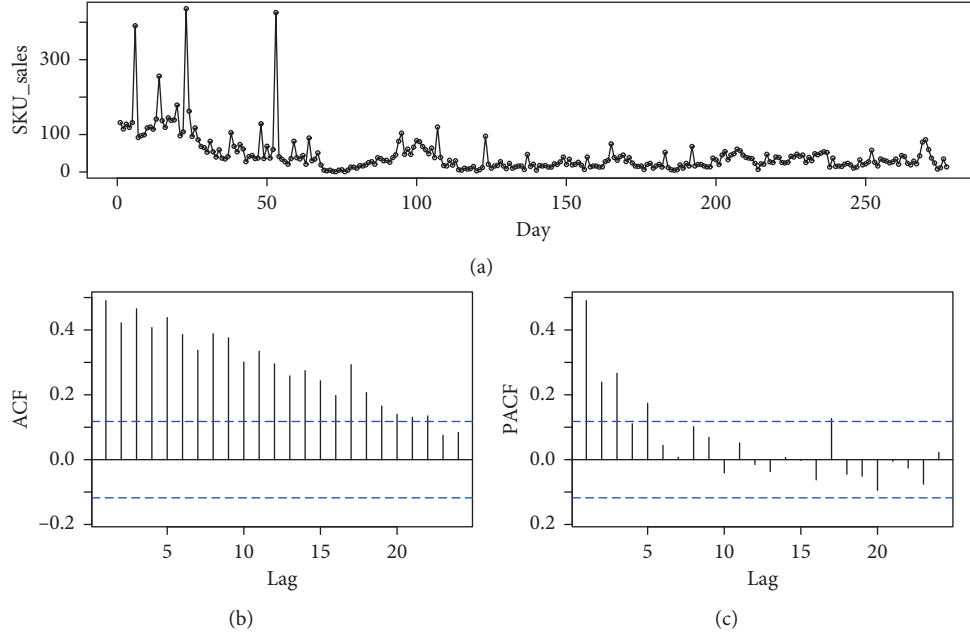
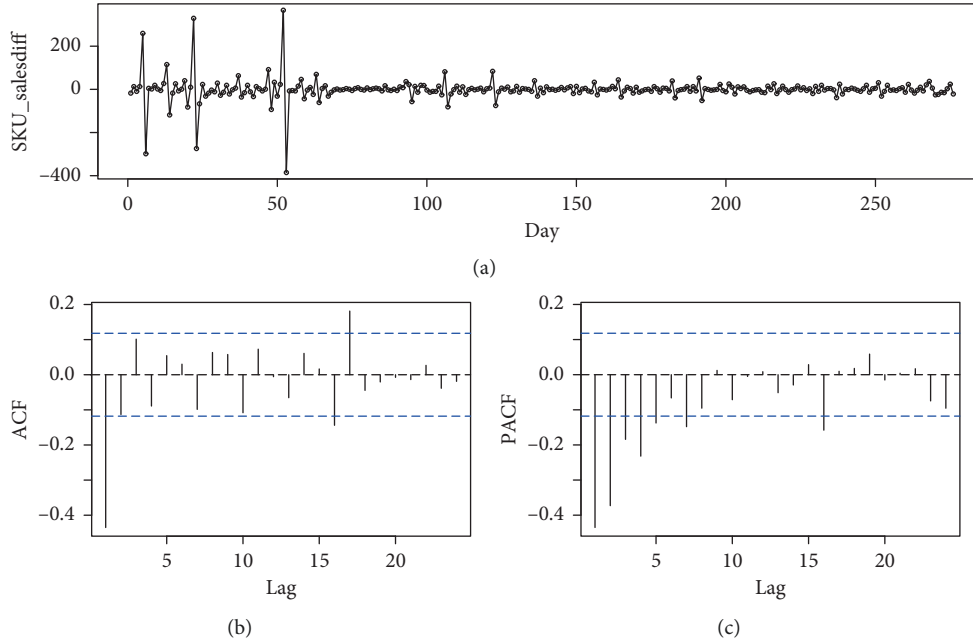FIGURE 9: Plots of (a) SKU sales with days change, (b) ACF, and (c) PACF.

FIGURE 10: Plots of (a) SKU sales with days change, (b) ACF, and (c) PACF after the first-order difference.

(3) Step 3. Calculate residuals of the optimal ARIMA: The prediction results from the 278th to the 381st day are obtained by using the trained ARIMA (2, 1, 4), denoted as ARIMA_forecast. Then, residuals between the prediction values ARIMA_forecast and the actual values SKU_sales are calculated, denoted as ARIMA_residuals.

(4) Step 4. Train A-XGBoost by setting ARIMA_residuals as the input and output: As shown in equation (14), the output data are composed of 8 columns of the matrix **R**, and the corresponding inputs are the residuals of the last 7 days (from Column 1 to 7):

$$
\mathbf{R} = \begin{bmatrix}
r_{271} & r_{272} & \cdots & r_{277} & r_{278} \\
r_{272} & r_{273} & \cdots & r_{278} & r_{279} \\
\vdots & \vdots & \cdots & \vdots & \vdots \\
r_{339} & r_{340} & \cdots & r_{345} & r_{346} \\
r_{340} & r_{341} & \cdots & r_{346} & r_{347}
\end{bmatrix}_{70 \times 8}.
\tag{14}
$$

TABLE 8: AIC values of the resulting ARIMA by auto.airma ( ) function.

ARIMA (p, d, q)	AIC	ARIMA (p, d, q)	AIC
ARIMA (2, 1, 2) with drift	2854.317	ARIMA (0, 1, 2) with drift	2852.403
ARIMA (0, 1, 0) with drift	2983.036	ARIMA (1, 1, 2) with drift	2852.172
ARIMA (1, 1, 0) with drift	2927.344	**ARIMA (0, 1, 1)**	**2850.212**
ARIMA (0, 1, 1) with drift	2851.296	ARIMA (1, 1, 1)	2851.586
ARIMA (0, 1, 0)	2981.024	ARIMA (0, 1, 2)	2851.460
ARIMA (1, 1, 1) with drift	2852.543	ARIMA (1, 1, 2)	2851.120

TABLE 9: AIC values and RMSE of ARIMA models under different parameters.

ARIMA model	ARIMA (p, d, q)	AIC	RMSE
1	ARIMA (0, 1, 1)	2850.170	41.814
2	ARIMA (2, 1, 2)	2852.980	41.572
3	ARIMA (2, 1, 3)	2854.940	41.567
4	**ARIMA (2, 1, 4)**	**2848.850**	**40.893**

TABLE 10: The performance evaluation of A-XGBoost.

A-XGBoost	Validation set	Test set
Minimum error	−0.003	−8.151
Maximum error	0.002	23.482
Mean error	0.000	1.213
Mean absolute error	0.001	4.566
Standard deviation	0.001	6.262
Linear correlation	1	−0.154
Occurrences	70	34

(5) Step 5. Calculate predicted residuals of the test set using the trained A-XGBoost in Step 4, denoted as $A-$ XGBoost_residuals: For the test set, the result of the 348th day is obtained by the setting ARIMA_residuals of the 341st–347th day as the input. Then, the result of the 349th day can be calculated by inputting ARIMA_residuals of the 342nd–347th day and $A-$ XGBoost_residuals of the 348th day into the trained A-XGBoost. The processes are repeated until the $A-$ XGBoost_residuals of the 349th–381st day are obtained.

(6) Step 6. Calculate the final prediction results: For the test set, calculate the final prediction results by summing over the corresponding values of $A-$ XGBoost_residuals and ARIMA_forecast, denoted as $A-$ XGBoost_forecast. The results of the A-XGBoost are summarized in Table 10.

4.3.3. C-A-XGBoost Model. The optimal combination weights are determined by minimizing the MSE in equation (6).

For the test set, the weights w_C and w_A are obtained based on the matrix operation equation (13) $\mathbf{W} = (\mathbf{B^T B})^{-1}\mathbf{B^T Y}$, where $w_C = 1.262$ and $w_A = 0.273$.

4.4. Models for Comparison. In this section, the following models are chosen for the comparison between the proposed models and other classical models:

ARIMA. As one of the common time series model, it is used to predict sales of time sequence, of which the processes are the same as the ARIMA in Section 4.3.2.

XGBoost. The XGBoost model is constructed and optimized by setting the selected features and the corresponding SKU sales as input and output.

C-XGBoost. Taking sales features of commodities into account, the XGBoost is used to forecast sales based on the resulting clusters by the two-step clustering model. The procedures are the same as that in Section 4.3.1.

A-XGBoost. The A-XGBoost is applied to revising residuals of the ARIMA. Namely, the ARIMA is firstly used to model the linear part of the time series, and then XGBoost is used to model the nonlinear part. The relevant processes are described in Section 4.3.2.

C-A-XGBoost. The model combines the advantages of C-XGBoost and A-XGBoost, of which the procedures are displayed in Section 4.3.3.

4.5. Results of Different Models. In this section, the test set is used to verify the superiority of the proposed C-A-XGBoost.

Figure 11 shows the curve of actual values SKU_sales and five fitting curves of predicted values from the 348th day to the 381st day, which is obtained by the ARIMA, XGBoost, C-XGBoost, A-XGBoost, and C-A-XGBoost.

It can be seen that C-A-XGBoost has the best fitting performance to the original value, as its fitting curve is the most similar in five fitting curves to the curve of actual values SKU sales.

To further illustrate the superiority of the proposed C-A-XGBoost, the evaluation indexes mentioned in Section 4.2.2 are applied to distinguishing the best model of the sales forecast. Table 11 provides a comparative summary of the indexes for the five models in Section 4.4.

According to Table 11, it can be concluded that the superiority of the proposed C-A-XGBoost is distinct compared with the other models, as its evaluation indexes are minimized.

C-XGBoost is inferior to C-A-XGBoost but outperforms the other three models, underlining that C-XGBoost is superior to the single XGBoost.

A-XGBoost has a superior performance relative to ARIMA, proving that XGBoost is effective for residual modification of ARIMA.

According to the analysis above, the proposed C-A-XGBoost has the best forecasting performance for sales of commodities in the cross-border e-commerce enterprise.
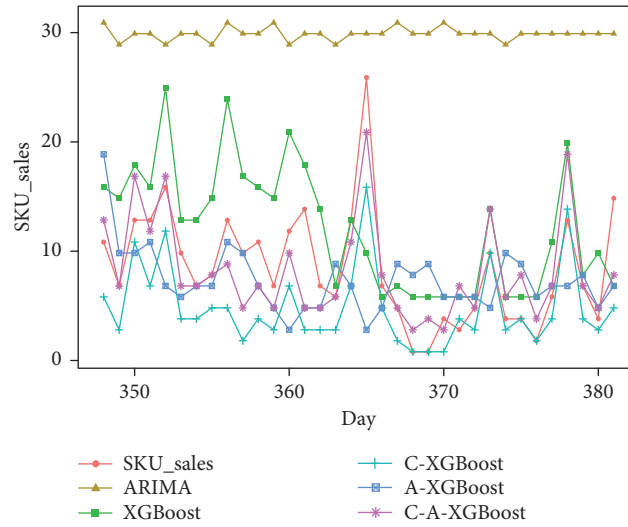
FIGURE 11: Comparison of the SKU sales with the predicted values of five models in Section 4.4. The *x*-axis represents the day. The *y*-axis represents the sales of SKU. The curves with different colors represent different models.

TABLE 11: The performance evaluation of ARIMA, XGBoost, A-XGBoost, C-XGBoost, and C-A-XGBoost.

Evaluation indexes	ARIMA	XGBoost	A-XGBoost	C-XGBoost	C-A-XGBoost
ME	−21.346	−3.588	1.213	3.647	0.288
MSE	480.980	36.588	39.532	23.353	10.769
RMSE	21.931	6.049	6.287	4.832	3.282
MAE	21.346	5.059	4.566	3.765	2.515

5. Conclusions and Future Directions

In this research, a new XGBoost-based forecasting model named C-A-XGBoost is proposed, which takes the sales features and tendency of data series into account.

The C-XGBoost is first presented combining the clustering and XGBoost, aiming at reflecting sales features of commodities into forecasting. The two-step clustering algorithm is applied to partitioning data series into different clusters based on selected features, which are used as the influencing factors for forecasting. After that, the corresponding C-XGBoost models are established for different clusters using the XGBoost.

The proposed A-XGBoost takes the advantages of the ARIMA in predicting the tendency of data series and overcomes the disadvantages of the ARIMA by applying the XGBoost to dealing with the nonlinear part of the data series. The optimal ARIMA is obtained in comparison of AICs under different parameters and then the trained ARIMA model is used to predict the linear part of the data series. For nonlinear part of data series, the rolling prediction is conducted by the trained XGBoost, of which the input and output are the resulting residuals by the ARIMA. The final results of the A-XGBoost are calculated by adding the predicted residuals by the XGBoost to the corresponding forecast values by the ARIMA.

In conclusion, the C-A-XGBoost is developed by assigning appropriate weights to the forecasting results of the C-XGBoost and A-XGBoost so as to take their respective strengths. Consequently, a linear combination of the two models' forecasting results is calculated as the final predictive values.

To verify the effectiveness of the proposed C-A-XGBoost, the ARIMA, XGBoost, C-XGBoost, and A-XGBoost are employed for comparison. Meanwhile, four common evaluation indexes, including ME, MSE, RMSE, and MAE, are utilized to check the forecasting performance of C-A-XGBoost. The experiment demonstrates that the C-A-XGBoost outperforms other models, indicating that C-A-XGBoost has provided theoretical support for sales forecast of the e-commerce company and can serve as a reference for selecting forecasting models. It is advisable for the e-commerce company to choose different forecasting models for different commodities instead of utilizing a single model.

The two potential extensions are put forward for future research. On the one hand, owing to the fact that there may be no model in which all evaluation indicators are minimal, which leads to the difficulty in choosing the optimal model. Therefore, a comprehensive evaluation index of forecasting performance will be constructed to overcome the difficulty. On the other hand, sales forecasting is actually used to optimize inventory management, so some relevant factors should be considered, including inventory cost, order lead time, delivery time, and transportation time.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

References

[1] Y. Jin, *Data Science in Supply Chain Management: Data-Related Influences on Demand Planning*, Proquest Llc, Ann Arbor, MI, USA, 2013.

[2] S. Akter and S. F. Wamba, "Big data analytics in e-commerce: a systematic review and agenda for future research," *Electronic Markets*, vol. 26, no. 2, pp. 173–194, 2016.

[3] J. L. Castle, M. P. Clements, and D. F. Hendry, "Forecasting by factors, by variables, by both or neither?," *Journal of Econometrics*, vol. 177, no. 2, pp. 305–319, 2013.

[4] A. Kawa, "Supply chains of cross-border e-commerce," in *Proceedings of the Advanced Topics in Intelligent Information and Database Systems*, Springer International Publishing, Kanazawa, Japan, April 2017.

[5] L. Song, T. Lv, X. Chen, and J. Gao, "Architecture of demand forecast for online retailers in China based on big data," in *Proceedings of the International Conference on Human-Centered Computing*, Springer, Colombo, Sri Lanka, January 2016.

[6] G. Iman, A. Ehsan, R. W. Gary, and A. Y. William, "An overview of energy demand forecasting methods published in 2005–2015," *Energy Systems*, vol. 8, no. 2, pp. 411–447, 2016.

[7] S. Gmbh, *Forecasting with Exponential Smoothing*, vol. 26, no. 1, Springer, Berlin, Germany, 2008.

[8] G. E. P. Box and G. M. Jenkins, "Time series analysis: forecasting and control," *Journal of Time*, vol. 31, no. 4, 303 pages, 2010.

[9] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[10] S. Ma, R. Fildes, and T. Huang, "Demand forecasting with high dimensional data: the case of SKU retail sales forecasting with intra- and inter-category promotional information," *European Journal of Operational Research*, vol. 249, no. 1, pp. 245–257, 2015.

[11] Ö. Gür Ali, S. Sayın, T. van Woensel, and J. Fransoo, "SKU demand forecasting in the presence of promotions," *Expert Systems with Applications*, vol. 36, no. 10, pp. 12340–12348, 2009.

[12] T. Huang, R. Fildes, and D. Soopramanien, "The value of competitive information in forecasting FMCG retail product sales and the variable selection problem," *European Journal of Operational Research*, vol. 237, no. 2, pp. 738–748, 2014.

[13] F. Cady, "Machine learning overview," in *The Data Science Handbook*, Wiley, Hoboken, NJ, USA, 2017.

[14] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny, "An empirical comparison of machine learning models for time series forecasting," *Econometric Reviews*, vol. 29, no. 5-6, pp. 594–621, 2010.

[15] A. P. Ansuj, M. E. Camargo, R. Radharamanan, and D. G. Petry, "Sales forecasting using time series and neural networks," *Computers & Industrial Engineering*, vol. 31, no. 1-2, pp. 421–424, 1996.

[16] I. Alon, M. Qi, and R. J. Sadowski, "Forecasting aggregate retail sales: a comparison of artificial neural networks and traditional methods," *Journal of Retailing and Consumer Services*, vol. 8, no. 3, pp. 147–156, 2001.

[17] G. Di Pillo, V. Latorre, S. Lucidi, and E. Procacci, "An application of support vector machines to sales forecasting under promotions," *4OR*, vol. 14, no. 3, pp. 309–325, 2016.

[18] L. Wang, H. Zou, J. Su, L. Li, and S. Chaudhry, "An ARIMA-ANN hybrid model for time series forecasting," *Systems Research and Behavioral Science*, vol. 30, no. 3, pp. 244–259, 2013.

[19] S. Ji, H. Yu, Y. Guo, and Z. Zhang, "Research on sales forecasting based on ARIMA and BP neural network combined model," in *Proceedings of the International Conference on Intelligent Information Processing*, ACM, Wuhan, China, December 2016.

[20] J. Y. Choi and B. Lee, "Combining LSTM network ensemble via adaptive weighting for improved time series forecasting," *Mathematical Problems in Engineering*, vol. 2018, Article ID 2470171, 8 pages, 2018.

[21] K. Zhao and C. Wang, "Sales forecast in e-commerce using the convolutional neural network," 2017, https://arxiv.org/abs/1708.07946.

[22] K. Bandara, P. Shi, C. Bergmeir, H. Hewamalage, Q. Tran, and B. Seaman, "Sales demand forecast in e-commerce using a long short-term memory neural network methodology," 2019, https://arxiv.org/abs/1901.04028.

[23] X. Luo, C. Jiang, W. Wang, Y. Xu, J.-H. Wang, and W. Zhao, "User behavior prediction in social networks using weighted extreme learning machine with distribution optimization," *Future Generation Computer Systems*, vol. 93, pp. 1023–1035, 2018.

[24] L. Xiong, S. Jiankun, W. Long et al., "Short-term wind speed forecasting via stacked extreme learning machine with generalized correntropy," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4963–4971, 2018.

[25] J. L. Zhao, H. Zhu, and S. Zheng, "What is the value of an online retailer sharing demand forecast information?," *Soft Computing*, vol. 22, no. 16, pp. 5419–5428, 2018.

[26] G. Kulkarni, P. K. Kannan, and W. Moe, "Using online search data to forecast new product sales," *Decision Support Systems*, vol. 52, no. 3, pp. 604–611, 2012.

[27] A. Roy, "A novel multivariate fuzzy time series based forecasting algorithm incorporating the effect of clustering on prediction," *Soft Computing*, vol. 20, no. 5, pp. 1991–2019, 2016.

[28] R. J. Kuo and K. C. Xue, "A decision support system for sales forecasting through fuzzy neural networks with asymmetric fuzzy weights," *Decision Support Systems*, vol. 24, no. 2, pp. 105–126, 1998.

[29] P.-C. Chang, C.-H. Liu, and C.-Y. Fan, "Data clustering and fuzzy neural network for sales forecasting: a case study in printed circuit board industry," *Knowledge-Based Systems*, vol. 22, no. 5, pp. 344–355, 2009.

[30] C.-J. Lu and Y.-W. Wang, "Combining independent component analysis and growing hierarchical self-organizing maps with support vector regression in product demand forecasting," *International Journal of Production Economics*, vol. 128, no. 2, pp. 603–613, 2010.

[31] C.-J. Lu and L.-J. Kao, "A clustering-based sales forecasting scheme by using extreme learning machine and ensembling linkage methods with applications to computer server,"

Engineering Applications of Artificial Intelligence, vol. 55, pp. 231–238, 2016.

[32] W. Dai, Y.-Y. Chuang, and C.-J. Lu, "A clustering-based sales forecasting scheme using support vector regression for computer server," *Procedia Manufacturing*, vol. 2, pp. 82–86, 2015.

[33] I. F. Chen and C. J. Lu, "Sales forecasting by combining clustering and machine-learning techniques for computer retailing," *Neural Computing and Applications*, vol. 28, no. 9, pp. 2633–2647, 2016.

[34] T. Chiu, D. P. Fang, J. Chen, Y. Wang, and C. Jeris, "A robust and scalable clustering algorithm for mixed type attributes in a large database environment," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2001.

[35] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: a new data clustering algorithm and its applications," *Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 141–182, 1997.

[36] L. Li, R. Situ, J. Gao, Z. Yang, and W. Liu, "A hybrid model combining convolutional neural network with XGBoost for predicting social media popularity," in *Proceedings of the 2017 ACM on Multimedia Conference—MM '17*, ACM, Mountain View, CA, USA, October 2017.

[37] J. Ke, H. Zheng, H. Yang, and X. Chen, "Short-term forecasting of passenger demand under on-demand ride services: a spatio-temporal deep learning approach," *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 591–608, 2017.

[38] K. Shimada, "Customer value creation in the information explosion era," in *Proceedings of the 2014 Symposium on VLSI Technology*, IEEE, Honolulu, HI, USA, June 2014.

[39] H. A. Abdelhafez, "Big data analytics: trends and case studies," in *Encyclopedia of Business Analytics & Optimization*, Association for Computing Machinery, New York, NY, USA, 2014.

[40] K. Kira and L. A. Rendell, "A practical approach to feature selection," *Machine Learning Proceedings*, vol. 48, no. 1, pp. 249–256, 1992.

[41] T. M. Khoshgoftaar, K. Gao, and L. A. Bullard, "A comparative study of filter-based and wrapper-based feature ranking techniques for software quality modeling," *International Journal of Reliability, Quality and Safety Engineering*, vol. 18, no. 4, pp. 341–364, 2011.

[42] M. A. Hall and L. A. Smith, "Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper," in *Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference. DBLP*, Orlando, FL, USA, May 1999.

[43] V. A. Huynh-Thu, Y. Saeys, L. Wehenkel, and P. Geurts, "Statistical interpretation of machine learning-based feature importance scores for biomarker discovery," *Bioinformatics*, vol. 28, no. 13, pp. 1766–1774, 2012.

[44] M. Sandri and P. Zuccolotto, "A bias correction algorithm for the Gini variable importance measure in classification trees," *Journal of Computational and Graphical Statistics*, vol. 17, no. 3, pp. 611–628, 2008.

[45] J. Brownlee, "Feature importance and feature selection with xgboost in python," 2016, https://machinelearningmastery.com.

[46] N. V. Chawla, S. Eschrich, and L. O. Hall, "Creating ensembles of classifiers," in *Proceedings of the IEEE International Conference on Data Mining*, IEEE Computer Society, San Jose, CA, USA, November-December 2001.

[47] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[48] A. Nagpal, A. Jatain, and D. Gaur, "Review based on data clustering algorithms," in *Proceedings of the IEEE Conference on Information & Communication Technologies*, Hainan, China, September 2013.

[49] Y. Wang, X. Ma, Y. Lao, and Y. Wang, "A fuzzy-based customer clustering approach with hierarchical structure for logistics network optimization," *Expert Systems with Applications*, vol. 41, no. 2, pp. 521–534, 2014.

[50] B. Wang, Y. Miao, H. Zhao, J. Jin, and Y. Chen, "A biclustering-based method for market segmentation using customer pain points," *Engineering Applications of Artificial Intelligence*, vol. 47, pp. 101–109, 2015.

[51] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems: Integrating Artificial, Intelligence and Database Technologies*, vol. 17, no. 2-3, pp. 107–145, 2001.

[52] R. W. Sembiring, J. M. Zain, and A. Embong, "A comparative agglomerative hierarchical clustering method to cluster implemented course," *Journal of Computing*, vol. 2, no. 12, 2010.

[53] M. Valipour, M. E. Banihabib, and S. M. R. Behbahani, "Comparison of the ARIMA and the auto-regressive artificial neural network models in forecasting the monthly inflow of the dam reservoir," *Journal of Hydrology*, vol. 476, 2013.

[54] E. Erdem and J. Shi, "Arma based approaches for forecasting the tuple of wind speed and direction," *Applied Energy*, vol. 88, no. 4, pp. 1405–1414, 2011.

[55] R. J. Hyndman, "Forecasting functions for time series and linear models," 2019, http://mirror.costar.sfu.ca/mirror/CRAN/web/packages/forecast/index.html.

[56] S. Aishwarya, "Build high-performance time series models using auto ARIMA in Python and R," 2018, https://www.analyticsvidhya.com/blog/2018/08/auto-arima-time-series-modeling-python-r/.

[57] J. H. Friedman, "Machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[58] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," 2016, https://arxiv.org/abs/1603.02754.

[59] A. Gómez-Ríos, J. Luengo, and F. Herrera, "A study on the noise label influence in boosting algorithms: AdaBoost, Gbm, and XGBoost," in *Proceedings of the International Conference on Hybrid Artificial Intelligence Systems*, Logroño, Spain, June 2017.

[60] J. Wang, C. Lou, R. Yu, J. Gao, and H. Di, "Research on hot micro-blog forecast based on XGBOOST and random forest," in *Proceedings of the 11th International Conference on Knowledge Science, Engineering and Management KSEM 2018*, pp. 350–360, Changchun, China, August 2018.

[61] C. Li, X. Zheng, Z. Yang, and L. Kuang, "Predicting short-term electricity demand by combining the advantages of ARMA and XGBoost in fog computing environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 5018053, 18 pages, 2018.

[62] A. M. Jain, "Complete guide to parameter tuning in XGBoost with codes in Python," 2016, https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/.