Soft Computing
https://doi.org/10.1007/s00500-023-08729-1

FOCUS

A novel weight-optimized LSTM for dynamic pricing solutions in e-commerce platforms based on customer buying behaviour

S. Suresh Kumar[1,2] · Martin Margala[2] · S. Siva Shankar[3] · Prasun Chakrabarti[4]

Accepted: 31 May 2023

Abstract

Dynamic pricing (DP) alters product prices at the ideal moment and under the ideal conditions to boost a company's profit. Numerous third-party DP solutions are currently available, offering APIs and standard regulations like competitive pricing. However, many organizations prefer to create their in-house solutions due to the complexity and size of enormous e-commerce (EC) operations and the limited customizability of that third-party software. This paper proposes a novel deep learning algorithm, namely weight-optimized long short-term memory (WOLSTM) for DP solutions based on customer buying or purchasing behaviour in EC platforms. The proposed work consists of five phases. The missing value imputation and normalization preprocessing technique was initially applied to the collected dataset. After that, the relevant features are extracted from the preprocessed dataset. Then, the dimensionality of the extracted feature set is reduced using the improved principal component analysis method. Next, similar customers are grouped using the hamming silhouette coefficient-based k-means approach, and finally, the prediction of DP for the grouped customers is made using WOLSTM. The proposed system's efficiency is analysed using the electronic products and pricing data dataset. The experimental results indicate that the proposed WOLSTM outperformed state-of-the-art methods regarding the classification metrics and the proposed one yields 98.99% accuracy with less RMSE of 2.356.

Keywords E-commerce · Customer buying behaviour · Dynamic pricing · Machine learning · Deep learning · Long short-term memory · Principal component analysis

1 Introduction

The Internet's explosive development significantly impacted the industry for fresh products. The Internet breaks the distance limit, allowing customers to purchase goods from anywhere in the world (Wang et al. 2022). Businesses have many opportunities to grasp their customers' online behaviour thanks to daily online activities (Ceci et al. 2014). EC can make purchasing easier for customers with a wide range of choices. It can also collect data on consumer consumption and create user profiles using big data technology (Xiao 2022; Zhu and Lin 2019). Online retail and online marketplaces are the fastest-growing sectors within the EC field, according to the PwC report on EC in India. About 80% of the goods sold through eTailing are books, clothes, accessories, and gadgets (Victor et al. 2018). Given the sheer volume of goods offered online, product pricing is quite challenging on EC platforms. Several variables influence product prices, so that pricing can

✉ S. Suresh Kumar
nice.ssk@gmail.com

Martin Margala
martin.margala@louisiana.edu

S. Siva Shankar
drsivashankars@gmail.com

Prasun Chakrabarti
drprasun.cse@gmail.com

1 Department of CSE, Swarnandhra College of Engineering and Technology, Narasapur, Andhra Pradesh, India

2 School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, USA

3 3Department of Computer Science and Engineering, KG Reddy College of Engineering and Technology, Hyderabad, India

4 ITM (SLS) Baroda University, Vadodara, India

Published online: 21 June 2023

fluctuate (Leung et al. 2019; Lou et al. 2021). According to market demand and supply capacity, dynamic pricing (Hernes et al. 2020) sells the same product to different customers or market groups at different rates to maximize revenue. This strategy has had noteworthy success in many retail industries. However, some customers have become aware of this occurrence and have reacted appropriately (Wu 2022) due to the different prices for the same product.

One of the most widely used modern types of pricing in EC is personalized or customized pricing, which means a dynamic price adjustment for consumers depending on the value that these customers attribute to the commodities (Chornous and Horbunova 2020). Because of heavy market competition, implementing pricing in the conventional mechanism of the sum of distribution costs and estimated profit can no longer be the seller's best action. As a result, companies that follow this strategy will see a decline in consumer loyalty and their ability to compete (Tong et al. 2020). Some research studies discovered that rising pricing is beneficial when customer demand for employees surpasses supply, which could attract more employees and, in the meantime, assign limited employees to customers who value them the most. In contrast, other research studies found that surge pricing triggers drivers in certain surge price zones to become inactive and cannot regulate supply and demand (Zhang and Wang 2021). In EC platforms, machine learning (ML) supports DP options. The airline industry is a well-known illustration of how ML is used in DP in EC, where the ticket cost is variable and greatly influenced by demand from the public. ML models, however, work better with simple linear data. Getting the best prediction outcomes can be difficult when the data are complex and have a nonlinear structure. Additionally, the forecast findings' accuracy decreases over time (Zhang and Wang 2021). The advancement of DL increases the effectiveness and speed of optimum price prediction (Guo 2022; Gao and Zhao 2022; Zhang 2021). However, because there needs to be a clear, comprehensive strategy for data mining in dynamic pricing for EC, it cannot fully exploit the effectiveness of DL. For dynamic pricing, this paper employs an optimized DL algorithm.

Understanding consumer behaviour necessitates pinpointing the variables that persuade people to purchase from websites, but doing so is challenging (Xiao et al. 2019; Safara 2022). A positive user experience will influence customer attitudes, boost purchase intentions, and encourage impulsive behaviour. As a result, it is essential to predict how the community user experience will affect consumers' buying intentions (Lu et al. 2021; Zhao and Shi 2022). It also helps to anticipate customer purchasing intentions to predict dynamic pricing accurately. Better pricing will be determined for a product using the customer's purchase choice as the basis. This is motivated to propose an optimized LSTM DL approach to address the dynamic pricing problem for online retailers based on customer buying behaviour. The main contributions of the work are as follows:

- To present efficient preprocessing schemes to standardize the dataset makes the prediction task more accessible and efficient.
- To propose an IPCA method as a dimensionality reduction (DR) scheme that diminishes the classifier's complexity and avoids misleading results.
- To propose an HSCKM algorithm for customer grouping to predict customer buying behaviour that helps to identify the purchase intentions of the customer based on some product features on a particular product.
- To present a WOLSTM approach for DP prediction that recommends the DP for similar customers based on their purchasing behaviour.
- To propose a modified shark smell optimization algorithm to tune the network parameters of LSTM that avoids the classifier's vanishing gradient and overfitting issues by enhancing the prediction outcomes.

The paper is organized as follows: Section 2 reviews studies on the existing work on customer buying behaviour prediction and DP solutions on the EC platform. Section 3 discusses the proposed work. Subsequently, the analysis results are presented and discussed in Sect. 4, while the conclusions and future works are drawn in Sect. 5.

2 Literature review

This section briefly reviews the recent methodologies for predicting the DP of the products in EC platforms. The survey regarding customers' purchasing behaviour predictions is also presented because they help the vendors to fix the prices for the particular customer at a particular time. The drawbacks identified in the current works are also discussed at the end of the survey, and the solutions to overcome the drawbacks are given.

Faehnle and Guidolin (2021) presented a DP prediction model for electronic market products using vector autoregressive processes (VAR) and lasso penalization (LP). The data for this pricing prediction were collected from openly available sources, which contained the pricing details of the latest smartphones in various brands such as Samsung, Apple and Huawei sold on EC platforms. The collected data were preprocessed, and the product's pricing was identified using VAR and LP. The model predicted the real-time price variations of the products in a single vendor based on the pricing variations of the product from the competitor, which will be helpful for EC companies to know their competitor's pricing strategies. Mohamed et al.

(2022) suggested ML and statistical approaches to predict the price of seasonal products. Initially, the data were collected from the publicly available dataset and partitioned into two sets for training and testing. Then, the ML models, such a support vector regression, random forest (RF), logistic regression, and ridge models and the statistical approach, namely autoregressive-integrated-moving average, were applied to the collected dataset for product price prediction. The relationships, patterns and critical features were captured using these models.

Namburu et al. (2022) presented a hybrid ML approach as a product pricing recommendation approach that helps vendors to price their products competitively by comparing competitors' pricing strategies on similar products. The method collected the dataset and extracted the features from the products. For pricing prediction, the extracted features were given to the hybrid ML classifiers such as LightBoost, XGBoost, and CatBoost. Kastius and Schlosser (2021) presented two different learning approaches, namely deep q networks (DQN) and soft actor-critic (SAC), for DP solutions of the products. The DQN was a q-learning model that utilized an artificial neural network to predict the product's DP based on its attributes. In contrast, SAC was the iteration-based policy gradient technique. The results showed that SAC performed better than DQN for product pricing. Chen et al. (2022) presented a price solution for the monopoly EC vendors using an information level of the customers. Consumer pricing was based on cross-network costs. Platforms for competitive online purchasing also sought to maximize profits. If the total number of consumers remained constant, the platforms would draw more merchants if customer information levels were higher. Due to the decreased bilateral user fees, platform earnings would also decrease. The system examined the effects of monopoly and competitive platforms adopting return measures to improve the level of consumer information on platform pricing from the viewpoint of consumer information level.

Esmeli et al. (2021) presented hybrid ML frameworks for customers' early purchase prediction among EC platforms. The analysis was performed on the YooChoose EC dataset that contained 6 months of session logs of various customers. The preprocessing and feature extraction phases were carried out for purchase behaviour prediction. Then, the hybrid ML classifiers such as RF, bagging, k-nearest neighbour (KNN), decision tree (DT), and naïve Bayes (NB) were utilized to detect the purchase intention of the customer. This analysis helped the vendors give better offers and discounts to their customers at their sessions. Chaudhuri et al. (2021) presented customers purchase behaviour model using a deep neural network. The system mainly used characteristics of customers and platform engagement to predict the customer's purchase intentions. Once collecting the data, preprocessing, such as missing value imputation and one hot encoding, was done to get the structured data from the dataset. The preprocessed data were given to the DNN for predicting the customer's purchase behaviour. Chaubey et al. (2022) utilized several ML classifiers' CPB predictions. The method utilized preprocessing scheme initially to structure the dataset for classification purposes. Then, it selected the essential features from the dataset using the Chi-square model for efficient classification. Finally, the ML classifiers include DT, KNN, RF, NB, artificial neural network, support vector machine (SVM), and stochastic gradient descent (SGD). The KNN and SVD showed better performance with full features in the dataset.

Alghanam et al. (2022) presented several classification models for customer purchase behaviour prediction. The real-time transactional data of the customers from the EC site were collected from Kaggle, and processing was performed initially to standardize the collected data. Then, the clustering concept was used to reduce the dataset size using k-means, decreasing the system's execution time. The clustered data were classified using CS-MC4, MLR, J48 and C4.5 for customer purchase prediction. The results showed that C4.5 and J48 attained higher classification accuracy than others. Xu et al. (2020) presented an ensemble learning model for user purchase behaviour prediction on EC platforms. The dataset's features indicated the customers' operations on their buying behaviours. The essential features were selected using the sort aggregation model to improve the prediction accuracy. Finally, the selected features were classified using a stacking integration framework that effectively predicted the purchase behaviour of online EC users.

2.1 Problem statement

The EC sector gathers customer data while using the network platform. EC businesses analyse customers and set prices based on the data given by the platform by assessing their consumption patterns. The recently developed works for CPB prediction and pricing predictions are listed above. However, some of the limitations are addressed in the surveyed works. Some works used ML algorithms for DP solutions (Faehnle and Guidolin 2021; Namburu et al. 2022), which produce better prediction results. However, as the amount of data increases, the system's efficiency declines due to the Ml algorithms' structural design. For instance, SVM and KNN are more appropriate for simple linear data. Getting the best prediction outcomes can be difficult when the data are complex and have a nonlinear structure. So author Kastius and Schlosser (2021) used deep reinforcement learning methods to predict optimal pricing. Nevertheless, deep reinforcement learning has a higher cost and complexity. Natural physical systems face severe restrictions on reinforcement learning due to the plague of dimensionality.

Additionally, the works Mohamed et al. (2022) and Chen et al. (2022) have greater time complexity due to the numerous calculations that must be made and could worsen with system scalability. Some works concentrate on CPB prediction because it helps to quickly find the optimal pricing in the EC platform (Esmeli et al. 2021; Chaudhuri et al. 2021; Chaubey et al. 2022; Alghanam et al. 2022; Xu et al. 2020). Here also, most works use ML algorithms for predicting customer buying behaviour (Esmeli et al. 2021; Chaubey et al. 2022; Alghanam et al. 2022), which is prone to errors. Consider developing an algorithm with data sets that are too tiny to be inclusive. This results in irrelevant information for the customers. In Chaudhuri et al. (2021), author uses a deep neural network, but it takes more training time because it directly takes the data from the dataset. However, the dataset has more relevant features and missing features. So the system takes more time to predict the customer buying behaviour and declines in performance accuracy. Also, Xu et al. (2020) uses the SE-stacking algorithm for customer buying behaviour prediction. However, there is no generally accepted configuration, and also, using the stacking technique is that stacking brings a lot of added complexity (Rajalakshmi and Minu 2014; Minu and Thyagarajan 2013).

The survey clearly shows that the already developed models for pricing solutions give satisfactory performance. However, they are limited with higher execution time, overfitting, and lower prediction accuracy because of the poor algorithmic design of the classifiers and the absence of effective feature selection and preprocessing models in pricing recommendations. So to overcome such drawbacks, this paper proposes an optimal DL model with an effective feature reduction technique that selects more appropriate features for pricing solutions in the EC platform with higher accuracy by updating the weights of the classifiers optimally (Simpson and Nagarajan 2021; Dhanalakshmi and Nagarajan 2020).

3 Research methodology

This paper proposes a novel DL model for DP solutions based on customer buying behaviour in EC platforms that group's customers based on their purchasing behaviour and fix appropriate pricing. The proposed system consists of five phases: data preprocessing, feature extraction, DR, customer grouping, and DP. Initially, the data collected from the dataset are preprocessed using missing value imputation and normalization. Then the attribute and the relevant features are extracted from the dataset. After extracting the features, the dimensionality is reduced using the IPCA method to increase the classifier accuracy. Then, the customer grouping is performed based on dimensionality-reduced features. Afterwards, the HSCKM algorithm groups the EC customers based on their purchasing behaviour. Then based on the customer grouping, the dynamic price range is determined using WOLSTM. Figure 1 describes the general flow of the proposed DP solutions according to the customer buying behaviour.

3.1 Data collection

The data for predicting the DP of the EC products based on the customer buying behaviour are collected from Kaggle, named EPPD, and it is available at https://www.kaggle.com/datasets/datafiniti/electronic-products-prices. The data are provided by Datafiniti's Product Database, which contains 15,000 electronic products with ten unique fields with pricing information. The database collects data from thousands of websites to generate standardized product, business, and property information databases. The dataset also contains brand, genre, merchant, source, and more information. The prediction of customer intentions and DP on EC platforms has extensively used this dataset.

3.2 Preprocessing

Data preprocessing is a procedure which explains how the selected data will be cleaned from all the noise or outliers. To reduce the execution time and enhance the final results, preprocessing of the data after data collection is recommended. Two preprocessing techniques are applied in the proposed system, which is explained as follows.

3.2.1 Missing value imputation

When gathering data, it is common to encounter missing values. Imputation is a standard method for processing missing data that assigns some values to the missing values in the dataset to build a complete data matrix for various procedures. It shows as 'NA', 'NaN', 'NULL', '0', 'Not applicable', and "None" in the dataset, and the existence of missing values diminishes the data available for analysis, limiting the statistical validity of the research and, ultimately, the dependability of its results. Furthermore, it introduces significant bias into the results and reduces data efficiency. The missing values in the dataset can be imputed using the mean imputation approach in this case. The missing values in this imputation are replaced with the mean value of the entire feature column. This is calculated by summing the values of each data point and dividing by the total number of data points in the set.

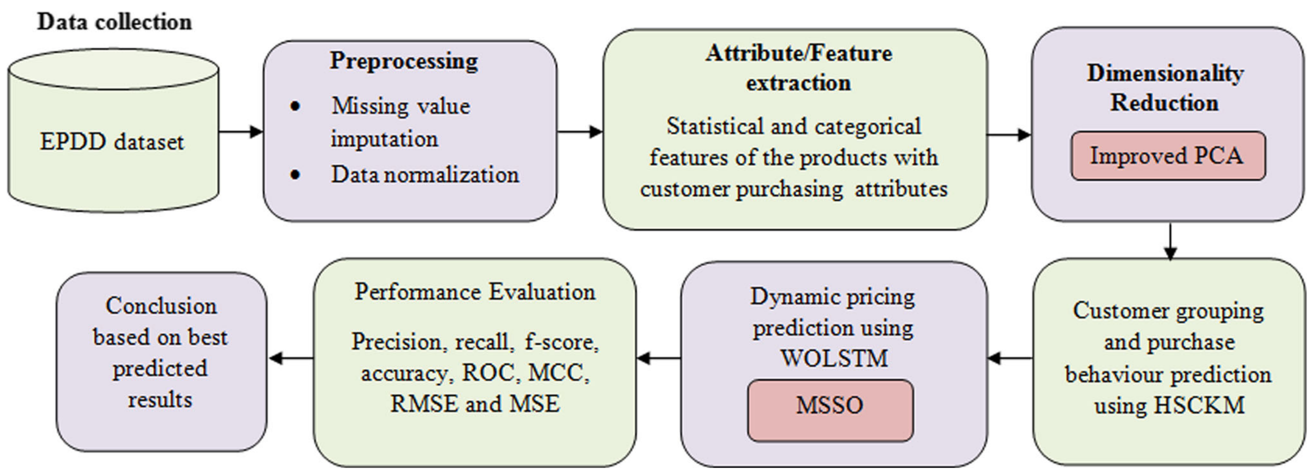

Fig. 1 Proposed flow diagram

3.2.2 Normalization

Data transformation, or normalization, transforms the original data into a different format, enabling efficient data processing. Data normalization's primary goal is to reduce or even eliminate redundant data. To achieve this, we normalize the data below using min–max normalization.

$$\tilde{N}\tilde{V}''_{\mathrm{val}} = \frac{\tilde{O}''_{\mathrm{val}} - \tilde{N}''_{o\min}\left(\tilde{N}''_{n\max} - \tilde{N}''_{n\min}\right) + \tilde{N}''_{n\min}}{\tilde{N}''_{o\max} - \tilde{N}''_{o\min}} \tag{1}$$

where $\tilde{N}\tilde{V}''_{\mathrm{val}}$ indicates the new normalized data, $\tilde{O}''_{\mathrm{val}}$ denotes the original value from the dataset, $\tilde{N}''_{o\min}$ and $\tilde{N}''_{o\max}$ refer to the old minimum and maximum value of the $\tilde{N}''_{n\max}$. The terms $\tilde{N}''_{n\min}$ and $\tilde{N}''_{n\max}$ represent the new minimum and maximum value of the $\tilde{N}''_{n\max}$. Here, we consider the desired range for normalization to be [0, 1], and therefore, Eq. (1) becomes,

$$\tilde{N}\tilde{V}''_{\mathrm{val}} = \frac{\tilde{O}''_{\mathrm{val}} - \tilde{N}''_{o\min}}{\tilde{N}''^{\,o\min}_{o\max}} \tag{2}$$

3.3 Feature extraction

Numerous real-world instances are provided for pricing and profit scenario strategies, which serve as the channel service's attributes. This stage includes extracting or choosing the attributes used for customer grouping and DP. The goal of feature extraction is to identify a minimal collection of attributes so that removing any unnecessary ones will not significantly reduce the usefulness of the resulting data. It lessens the computational complexity of learning and prediction algorithms and reduces the computational expense incurred when measuring unimportant characteristics. Table 1 lists the attributes that are extracted from the dataset for DP solutions.

These attributes include both statistical and categorical information about the products. Statistical features of the dataset can be defined and calculated via statistical analysis, and some of the features may be discrete values that are not in an ordered relationship. Thus it is known as a categorical feature. The statistics typically follow similar pricing strategies when the features belong to the same category. Grouping similar products improves prediction accuracy. Tables 2 and 3 show examples of the dataset's extracted statistical and categorical features.

Along with the above features, the features such as the purchase by category (PCT), purchase by the offer (POR), purchase by the company (PCY), purchase by channel (PCN), purchase by quantity (PQT), purchase by brand (PBD), and the purchase amount of each customer are extracted to make the prediction of purchase behaviour and DP stronger.

3.4 Dimensionality reduction

DR is done after extracting features from the dataset. It transforms the higher dimension of features into a lower dimension. The DR process removes the redundant data from the extracted feature set. It transforms the data into a lower dimension that helps to improve the prediction accuracy of the classification process and diminishes the classifier's training time. Also, a system's complexity can be decreased through DR, preventing the network from overfitting. Here, the DR is performed using the improved principal component analysis (IPCA) method. PCA is a popular method of DR that minimizes information loss while improving interpretability. It does so by generating new uncorrelated variables that successively maximize variance. However, when using PCA, the features become

Table 1 Dataset attributes

Name	Descriptions
Product	Product name
Product brand	Brand of the product
Product category	Different browsing categories of the product
Product subcategory 1	The product's subcategory at one level deep
Product subcategory 2	The product's specific category at two levels deep
Selling price	Product's selling price at a specific time or date
Selling date	The date at which the product was sold at the specific price
Product rating	Consumers' rating for the product

Table 2 Extracted statistical features

S. no.	Product Id	Product brand Id	Product category	Subcategory 1	Subcategory 2	Product rating
1	P-4452	B-3078	Jewellery	Necklaces chains	Necklaces	3.9
2	P-2453	B-3078	Jewellery	Bangles bracelets armlets	Bracelets	3.0
3	P-6802	B-1810	Apparels/clothing	Women's clothing	Western wear	1.5
	P-8454	B-3078	Apparels/clothing	Women's clothing	Western wear	1.4
5	P-2610	B-659	Computers	Network components	Routers	4.3

Table 3 Extracted categorical features

S. no.	Product	Product brand	Product category	Subcategory 1	Subcategory 2	Product rating
1	P-2610	B-659	9	11	159	4.3
2	P-2453	B-3078	17	139	387	3.0
3	P-6802	B-1810	38	119	118	1.5
4	P-4452	B-3078	12	40	155	3.9
5	P-8454	B-3078	17	86	344	1.4

less interpretable because these principal components cannot be read or understood. So, in data standardization, the categorical variables must convert to numerical form. Thus, the proposed PCA uses a binary encoding method for data standardization. Also, conventional PCA evaluates a correlation between features based on covariance calculation. However, the covariance calculation has the drawback of only measuring the affiliation between the data, whereas the data strength measurement needs to be performed. Nevertheless, strength analysis is effective for reducing the dimension. Therefore, this PCA used Gaussian kernel (GK) instead of covariance estimation to overcome this downside. This kernel estimation implicitly maps the data into a lower-dimensional space. These enhancements in conventional PCA are named improved PCA (IPCA). The steps involved in IPCA are explained as follows:

Step 1 Data standardization.

The extracted features of the products are multi-dimensional and diverse, so they need to be standardized before any further treatment. Let $(\tilde{E}\tilde{F}''_{\text{set}})$ be the extracted feature set from the dataset that are standardized using binary encoding. This scheme first converts categorical feature value into a numerical value, then the number is transformed into a binary number, and the binary value is split into different columns.

Step 2 After standardizing the features, the GK between each feature is computed. GK guarantees a globally optimal predictor which minimizes both the estimation and approximation errors of a DR. It is computed using the following equation,

$$\tilde{G}\tilde{K}''_n\left(\tilde{S}''_{\text{fs}}\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left(\frac{\tilde{S}''_{\text{fs}}-\mu}{2\sigma^2}\right)} \tag{3}$$

where $\tilde{S}''_{\text{fs}}$ indicates the standardized features, $\tilde{G}\tilde{K}''_n$ refers to the GK function, σ and μ represent the standard deviation and mean.

Step 3 Compute the eigenvector (EVT) and eigenvalue (EV). The EVT is a nonzero matrix that changes at most by a scalar factor when the linear transformation is applied to it. The corresponding EV is a factor by which the EVT is scaled. The EVT matrix (ψ) for each $\left(\tilde{G}\tilde{K}''_n\left(\tilde{S}''_{\text{fs}}\right)\right)$ is modelled as,

$$\psi = \begin{bmatrix} \nabla^{1,1} & \nabla^{1,2} & \nabla^{1,n} \\ \nabla^{2,1} & \nabla^{2,2} & \nabla^{2,n} \\ \nabla^{n,1} & \nabla^{n,2} & \nabla^{n,n} \end{bmatrix} \tag{4}$$

Here, $\nabla^{n,n}$ refers to the EVTs of each $\left(\tilde{G}\tilde{K}''_n\left(\tilde{S}''_{\mathrm{fs}}\right)\right)$.

Step 4 Further, the EVs (ξ) are evaluated using the below equation,

$$\tilde{G}\tilde{K}''_n\left(\tilde{S}''_{\mathrm{fs}}\right) * \psi = \psi * \xi \tag{5}$$

Step 5 Once the EVTs and EVs are estimated, they are arranged in descending order, with the EVT with the highest EV being one of the most important and formulating the principal component. Thus, the principal components of minor importance can be excluded in order to reduce the data dimension. Thus, the dimensionality-reduced input $\tilde{S}''_{\mathrm{fs}}$ becomes $\tilde{d}_i$—in a lower dimensional subspace. These data are then given to the customer grouping process.

3.5 Customer grouping

Customer grouping is performed after the DR. It separates the customers into several small groups based on their buying or purchasing behaviours. Customer behaviour indicates the customer's behaviour toward a specific product, such as customer views, likes, dislikes, ratings and reviews, purchases, and wish lists. This helps separate concerns, which in turn helps us DP. Based on these behaviours, similar customers are grouped using the HSCKM algorithm. K-Means (KM) clustering algorithm is a famous prototype-based clustering method that groups similar data points in data set by randomly selecting centroid points. It is relatively faster and functions well on large datasets than other clustering techniques. In addition, it diminishes the misclassification rate of data in customer grouping. Similar data points are grouped in the conventional KM algorithm using Euclidean distance. However, if two data vectors have no attribute values in common, they may have a smaller distance than the other pair of data vectors containing the same attribute values.

So the proposed system uses Hamming distance (HD) that effectively computes the distance between the data points even if the data points do not contain any common attribute. So the problem of assigning small values to the non-similar data can be avoided. Moreover, the KM algorithm's goodness is computed using Silhouette coefficient (SC). It analyses how well the resulting clusters are separated. These modifications in conventional KM algorithms for improving the performance are named Hamming Silhouette coefficient-based K-MEANS (HSCKM) algorithm. The steps of HSCKM are given as follows:

Step 1 Initially select *f*—reduced features randomly as cluster centres $\tilde{G}''_{\mathrm{ce}} = \{\tilde{g}_1, \tilde{g}_2, \tilde{g}_3, \ldots \tilde{g}_f\}$, from the dimensionality-reduced feature dataset $\tilde{R}''_{\mathrm{ce}} = \{\tilde{r}_1, \tilde{r}_2, \tilde{r}_3, \ldots \tilde{r}_f\}$.

Step 2 For each reduced feature $\tilde{R}''_{\mathrm{ce}}$, find the distance using HD (HamDist) with respect to the cluster centre $\tilde{G}''_{\mathrm{ce}}$. The HD is the number of features at which the corresponding cluster centre in the two points is different. It improves the clustering accuracy by accurate grouping. It is expressed as follows:

$$\mathrm{HamDist}\left(\tilde{R}''_{\mathrm{ce}}, \tilde{G}''_{\mathrm{ce}}\right) = \sum_{n=o}^{k} \left(\tilde{R}''_{\mathrm{ce}}\right)_n \oplus \left(\tilde{G}''_{\mathrm{ce}}\right)_n \tag{6}$$

where $\oplus$ indicates the logical exclusive symbol or XOR operation. Based on the above-mentioned equation, the dimensionality-reduced features are assigned to the nearest cluster centroid $\left(\tilde{G}''_{\mathrm{ce}}\right)$.

Step 3 Compute the mean value of each $\tilde{G}''_{\mathrm{ce}}$ and the, respectively, assigned reduced feature.

Step 4 Follow steps 2 and 3 again until the cluster centre mean value from this iteration is the same as the one from the previous iteration.

Step 5 Compute the SC once the grouping of the customers is done. The output of the coefficient is located between [− 1, + 1]. If the value is close to + 1, the objects (customers) are grouped far away from adjacent clusters. In contrast, if it is − 1, the data preprocessing may have been incorrect, or the objects (customers) may have been allocated to the incorrect cluster. It is said in the following way:

$$\tilde{S}''_c = \frac{\left((\mathrm{MI})_c - (\mathrm{ME})_c\right)}{\max\left((\mathrm{ME})_c,\ (\mathrm{MI})_c\right)} \tag{7}$$

where $\tilde{S}''_c$ indicates the SC, $(\mathrm{ME})_c$ refers to the mean value of intra-distance cluster to all objects in a cluster and $(\mathrm{MI})_c$ denotes the minimum average distance from *c*-th clusters to other clusters. After determining the SC, take the cluster with the highest silhouette value as per the evaluation method.

3.6 Dynamic pricing

Once the customers are grouped according to their purchasing behaviour, DP prediction is made for the same group of customers using WOLSTM. Long short-term memory (LSTM) is a deep learning (DL) architecture based on a recurrent neural network (RNN). DL is a subset of ML, a three- or more-layered neural network. These neural networks are inspired by the shape and function of biological neurons in the human brain and are meant to learn from massive amounts of data. Deep learning systems can learn and improve automatically from data without manual feature engineering. Deep learning has had substantial success in various sectors, and its application is projected to expand as more data becomes available and more powerful computing resources become available. LSTMs are particularly useful for dynamic pricing tasks because they can learn long-term dependencies in data. Time-series forecasting models can use LSTM to estimate future values based on earlier sequential information. This increases the accuracy of demand forecasters, resulting in better business decisions. LSTM networks address the issue of vanishing gradients or long-term reliance on RNNs. Gradient disappearing is data loss in a neural network as connections reoccur over time. Simply put, LSTM addresses vanishing gradients by disregarding worthless information or data in the network. The input, forget, and output gates in LSTM control the flow of the input data, its storage, and output. Each gate is a separate neural network and can be considered a filter. It solves the long-term dependency issue and performs remarkably well in DP. However, the LSTM training methods have the following drawbacks: gradient descent can become trapped in local minima with the random selection of the network's weight and bias, which results in a slower convergence issue and makes it impossible to identify the error function's global minimum. The classifier's random weight and bias values are optimally selected to overcome these shortcomings using a modified shark smell optimization (MSSO) technique that prevents the network from falling into vanishing gradient problems by minimizing classification errors and enhancing accuracy. The below steps explain the working of LSTM for DP, and its diagrammatic representation is shown in Fig. 2.

The LSTM network consists of the memory cell and three multiplicative gates such as an input $\left(\tilde{IG}_s''\right)$, forget $\left(\tilde{FG}_s''\right)$ and output gates $\left(\tilde{OG}_s''\right)$. Each gate enables continuous processes for the cells, and there are frequent relationships between the cells. The input data can be stored in the cell's state of the network if the input gate allows it.

$$\tilde{FG}_s'' = \mu\left(\omega_{\tilde{FG}''}\ \left(\tilde{h}_{s-1}, \tilde{y}_s\right) + \tilde{B}_{\tilde{FG}''}\right) \tag{8}$$

$$\tilde{IG}_s'' = \mu\left(\omega_{\tilde{IG}''}\ \left(\tilde{h}_{s-1}, \tilde{y}_s\right) + \tilde{B}_{\tilde{IG}''}\right) \tag{9}$$

$$\tilde{OG}_s'' = \mu\left(\omega_{\tilde{OG}''}\ \left(\tilde{h}_{s-1}, \tilde{y}_s\right) + \tilde{B}_{\tilde{OG}''}\right) \tag{10}$$

where μ refers to the sigmoid function, $\omega_{\tilde{FG}''}, \omega_{\tilde{IG}''}$, and $\omega_{\tilde{OG}''}$ represent the weight matrices of $\tilde{FG}_s'', \tilde{IG}_s''$, and $\tilde{OG}_s''$ respectively, $\tilde{B}_{\tilde{FG}''}, \tilde{B}_{\tilde{IG}''}$, and $\tilde{B}_{\tilde{OG}''}$ refer to the bias matrices of $\tilde{FG}_s'', \tilde{IG}_s''$, and $\tilde{OG}_s''$ respectively. The preceding output at $(t-1)$th timestamp is indicated as $\tilde{h}_{s-1}$, and $\tilde{y}_s$ signifies the current input vector at sth time stamp. In general, the weight and bias of the LSTM network are selected at random between 0 to $n-1$. In the proposed system, these values are chosen optimally using MSSO algorithm to improve the classification accuracy and avoid misclassification errors in pricing prediction.

Shark smell optimization (SSO) is a stochastic search optimization method that starts with a set of random solutions and then searches for the best optimal individuals at the end of the iteration. Shark is the best hunters in nature because of their ability to discover their prey in minimal time. The shark discovers the prey in an ample search space based on its excellent smelling sense. If the prey is injured, the sharks move towards that prey based on the prey's blood smell injected in the water particles. However, in traditional SSO algorithms, population initialization is done randomly and converges too early, resulting in suboptimal performance. So to prevent premature convergence, the population is initialized using the oppositional learning technique. In addition to balancing the exploration and exploitation capabilities of the algorithm, an inertia weight is adopted in the position updating phase of the SSO, which avoids the problem of the local optimal solutions when performing a global search of the algorithm. These modifications in conventional SSO for its performance improvement are named MSSO. The steps involved in the MSSO algorithm are explained as follows:

The algorithm starts with the population initialization phase. In this step, the shark's location must be computed at the initial point using the oppositional learning method. An opposite learning-based initialization offers a higher probability of a global optimum solution than the random initialization of the population. It is expressed as follows:

$$\hat{U}\hat{O}_x = \left[\hat{u}\hat{o}_x^1, \hat{u}\hat{o}_x^2, \hat{u}\hat{o}_x^3, \ldots, \hat{u}\hat{o}_x^j\right] \tag{11}$$

where $\hat{U}\hat{O}_x = \hat{L}\hat{V}_x + \lambda_x - \varphi_x$ with $\hat{U}\hat{O}_x \in \left[\hat{L}\hat{V}_x, \lambda_x\right]$ refers to the position of xth low variance blocks $\hat{U}\hat{O}_x$ in the hth dimension of oppositional blocks and φ_x indicates the xth location of low variance blocks. After population initialization, the fitness of the individuals is computed based on the classifier's accuracy. It is expressed as follows:

$$\overset{\leftrightarrow}{F}_{\text{fitfn}} = \text{Max}\left(\text{acc}_{\text{ps}}''\right) \tag{12}$$

where $\left(\text{acc}_{\text{ps}}''\right)$ refers to the accuracy. Divide the number of two accurate predictions by the total number of samples to calculate accuracy, which is expressed as

$$\left(\text{acc}_{\text{ps}}''\right) = \frac{T_{\text{tpos}}'' + T_{\text{tne}}''}{T_{\text{numsam}}''} \tag{13}$$

where T_{numsam}'' indicates the total number of samples, T_{tpos}'' and T_{tne}'' indicate true positive and true negative,

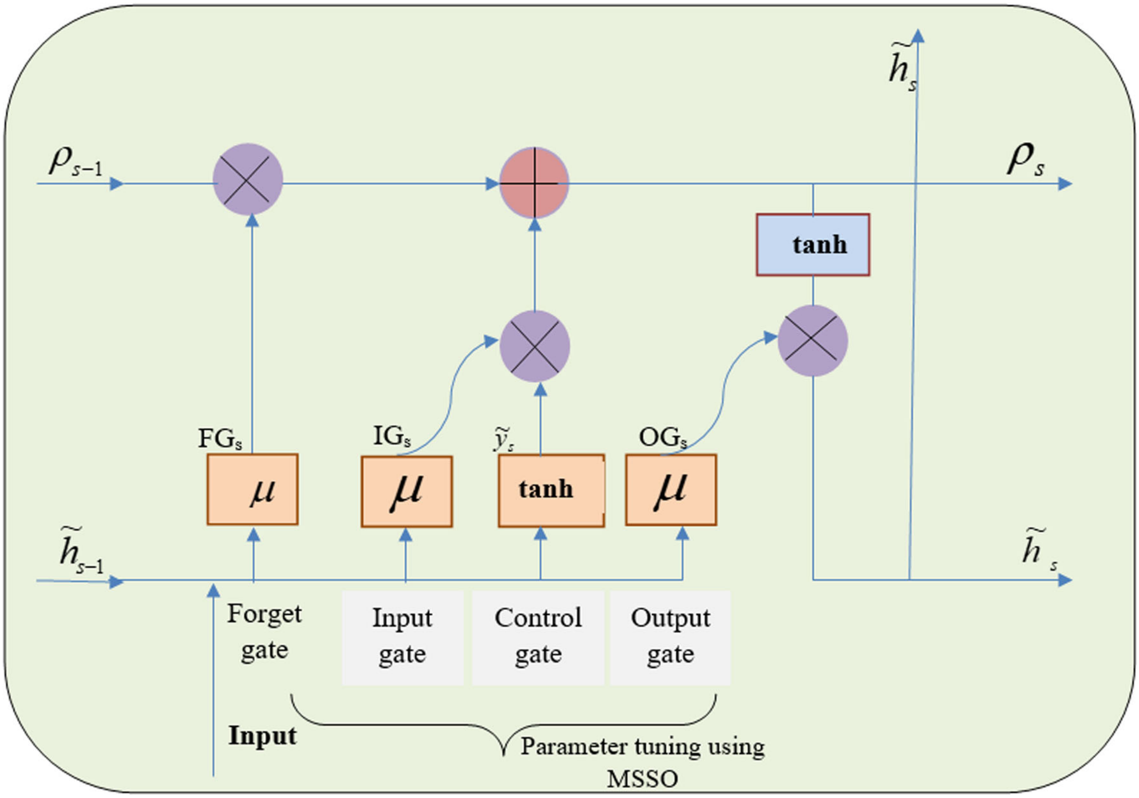

Fig. 2 Structure of LSTM model

respectively. The individuals attaining higher fitness are chosen as the best in the current iteration for selecting weights and biases. The higher fitness of the individuals indicates the shark's closest position towards the prey. The sharks move faster towards the injured prey, which releases a more intense scent in the water. So the initial velocity of the prey is expressed as

$$\left[\overleftrightarrow{V}''_1, \overleftrightarrow{V}''_2, \ldots \overleftrightarrow{V}''_{npv}\right] \tag{14}$$

The velocity in each dimension is denoted as

$$\overleftrightarrow{V}^t_x = \kappa_t \cdot \delta_1 \cdot \nabla\left(\overleftrightarrow{F}_{\text{fitfn}}\right)|\hat{U}\hat{O}^t_x, \quad a = 1,2,\ldots,npv,\ t = 1,2,\ldots t_{\max} \tag{15}$$

where $\overleftrightarrow{V}^t_x$ indicates the shark's velocity, $\overleftrightarrow{F}_{\text{fitfn}}$ denotes the fitness function, ∇ refers to the gradient of the fitness function at the position $\hat{U}\hat{O}^t_x$, $t_{\max}$ signifies the maximum number of stages for the shark's forward movement, t—refers to the number of stages, npv is the number of velocity vectors, κ_t indicates a value between [0, 1], and δ_1 proffers a uniformly distributed random variable between [0,1]. So the new position $\overleftrightarrow{Z}^{t+1}_x$ of the shark is computed based on its previous position and speed, and that can be expressed as:

$$\overleftrightarrow{Z}^{t+1}_x = \hat{U}\hat{O}^t_x + \overleftrightarrow{V}^t_x \cdot \Delta\alpha_t \cdot \xi_{\text{weight}} \tag{16}$$

where $\Delta\alpha_t$ indicates the time interval of the tth iteration, ξ_{weight} refers to the inertia weight that is computed using Eq. (17). In the earlier optimization phase, more significant inertia weight utilization maintains the algorithm's diversity and improves the global search capability of the algorithm. In later phases of iteration, choosing the smaller inertia weight results in better convergence and enhanced local search ability.

$$\zeta_{\text{weight}} = \left(\zeta \min_{\max}^{e^{(-t/T)}} + \zeta_{\min}\right) \tag{17}$$

where $\zeta \max_{\min}$ refers to the inertia weight's minimum and maximum value and t and T denote the current and maximum iteration number. Then, the sharks initiate the local search process to discover better candidate solutions and that can be performed using Eq. (8):

$$\overleftrightarrow{N}^{t+1,e}_x = \overleftrightarrow{Z}^{t+1}_x + \delta_2 \cdot \overleftrightarrow{Z}^{t+1}_x \tag{18}$$

where δ_2 indicates a uniformly distributed random value between [0, 1], $e = 1,2,3,\ldots E$ refers to the number of points at each stage of the location search. The shark moves towards the prey when it discovers a stronger smelling point in the rotation search. The location search formula is described as follows,

$$\hat{U}\hat{O}_x^{t+1} = \arg\max\left\{ \underset{\text{fitfn}}{\overleftrightarrow{F}}\left(\overleftrightarrow{Z}_x^{t+1}\right), \underset{\text{fitfn}}{\overleftrightarrow{F}}\left(\overleftrightarrow{N}_x^{t+1,1}\right), \ldots, \underset{\text{fitfn}}{\overleftrightarrow{F}}\left(\overleftrightarrow{N}_x^{t+1,E}\right) \right\} \tag{19}$$

where $\overleftrightarrow{Z}_x^{t+1}$ is attained through the linear movement and $\overleftrightarrow{N}_x^{t+1,E}$ is attained through rotation movement. The sharks will choose its next location $\hat{U}\hat{O}_x^{t+1}$ according to the higher evaluation index of the candidate solution. The above computations are performed on the weight and bias values for all three gates, and the optimal ones are selected to tune the network in DP prediction. The pseudocode for the MSSO algorithm is shown in Fig. 3.

After weight and bias optimization, the cell state, candidate cell state and the final output are computed as,

$$\underset{s}{\overleftrightarrow{\rho}} = \tanh\left(\varpi_\rho\left(\tilde{h}_{s-1}, \tilde{y}_s\right) + \tilde{B}_\rho\right) \tag{20}$$

$$\rho_s = \tilde{F}\tilde{G}_s'' * \rho_{s-1} + \tilde{I}\tilde{G}_s'' * \rho_s \tag{21}$$

$$\tilde{h}_s = \tilde{O}\tilde{G}_s'' * \tanh(\rho_s) \tag{22}$$

where ρ_s and ρ_{s-1} indicate the new and preceding cell states at the timestamps of s and also $s-1$. The term $\underset{s}{\overleftrightarrow{\rho}}$ denotes an output of tanh that is a candidate cell state at s, and * refers to the element-wise multiplication of the vectors. Once the price range for any customer group is depicted, the process will be terminated, and an offer value will be given to the customer.

4 Results and discussion

Here the experimental outcomes obtained by the proposed method for DP recommendation are analysed with the existing related schemes regarding classification performance metrics. The data for executing the proposed method are obtained from the publicly available dataset, and the proposed method is implemented in PYTHON with the following machine configurations: Intel(R) Core(TM) i5-7200U CPU@ 2.50 GHz with 64-bit Windows 10 OS. The dataset used for analysing the proposed method is EPDD, in which 80% of the data are used for training and 20% for testing. The existing techniques taken for comparison are LSTM, RNN, deep belief network (DBN), and convolutional neural network (CNN). These methods are compared with the proposed WOLSTM regarding precision (PN), recall (RL), f-measure (FE), accuracy (AY), receiver operating characteristics (ROC), Matthews correlation coefficient (MCC), mean absolute error (MAE), root-mean-square error (RMSE) and mean absoluter percentage error (MAPE). Table 4 shows the classifiers' results for predicting DP detection metrics such as PN, RL, FE, and AY.

The metrics such as PN, RL, FE, and AY are the major ones to detect the performance of any classification model used in the detection process, which should be high to prove the accurate detection of the models. Table 4 clearly shows that the proposed WOLSTM attains the best results for all detection metrics compared to existing schemes. The existing LSTM, DBN, CNN, and RNN attain the AY of 96.87%, 94.56%, 93.79%, and 91.68%, respectively, but the proposed one attains 98.99% of prediction AY, which is higher than the existing schemes. Likewise, the proposed

Fig. 3 Pseudocode of the MSSO algorithm

Input: Random weight values for an input
Output: Optimal weight and bias value

Begin
  Initialize the SSO population using opposition method using equation (11)
  Set npv, and κ_t
  Evaluate the fitness of each SSO using equation (12)
  Define initial velocity vectors as $\left[\vec{V}_1'', \vec{V}_2'', \ldots\ldots\ldots\ldots \vec{V}_{npv}''\right]$
  For each $t = 1 : t_{\max}$ **do**
    Update the new position by using equation (16)
    Update rotational search formula for shark position using equation (17)
    Update the location search by using
    $$\hat{U}\hat{O}_x^{t+1} = \arg\max\left\{\vec{F}_{fitfn}\left(\vec{Z}_x^{t+1}\right), \vec{F}_{fitfn}\left(\vec{N}_x^{t+1,1}\right), \ldots\ldots\ldots\ldots, \vec{F}_{fitfn}\left(\vec{N}_x^{t+1,E}\right)\right\}$$
  End for
  Return the optimal weight values and biases based on fitness
End

Table 4 Comparison of proposed WOLSTM and existing classifiers

Metrics (%)/techniques	Proposed WOLSTM	LSTM	DBN	CNN	RNN
AY	98.99	96.87	94.56	93.79	91.68
PN	98.71	96.17	94.16	93.29	91.45
RL	99.09	96.93	94.66	93.89	91.89
FE	98.92	96.68	94.35	93.67	91.61

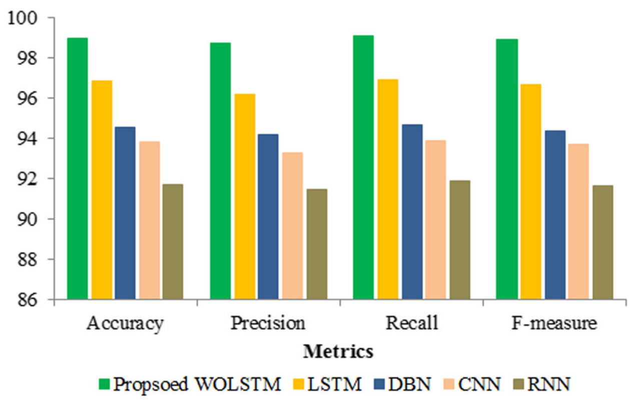

Fig. 4 Analysis of classifiers regarding detection metrics

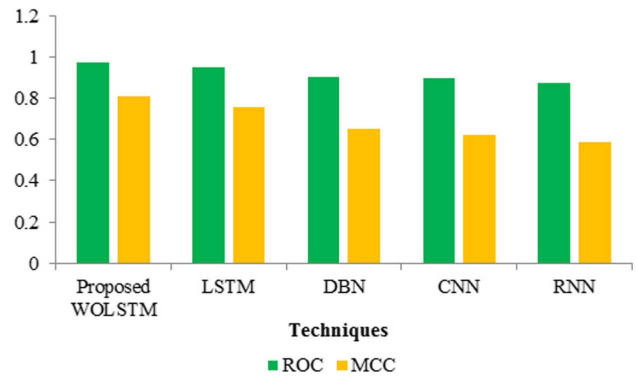

Fig. 5 ROC and MCC analysis

method attains higher values (98.71% of PN, 99.09% of RL, and 98.92% of AY) for other detection metrics. Thus, the proposed one is more accurate than conventional methodologies for DP recommendation. A diagrammatic representation of Table 4 is shown in Fig. 4.

Then, the comparison of the classifiers is made based on ROC and MCC. ROC compares two operating characteristics, and this curve is drawn between the true-positive and false-positive rates to show the performance efficiency of a classifier. The MCC of the classifier ranges between [− 1, 1]. The model obtains a good score in MCC only if it attains the best results in all four confusion matrices such as true positive, true negative, false positive and false positive and false negative. Figure 5 shows that the proposed one performs better than the conventional methods. The proposed WOLSTM achieves ROC and MCC of 0.976 and 0.812, respectively, but the existing LSTM, DBN, CNN, and RNN offer ROC of 0.953, 0.902, 0.898, and 0.876, respectively, and MCC of 0.756, 0.654, 0.621, and 0.587, respectively, which is lower than the proposed one. Thus, the results indicate that the proposed WOLSTM outperforms the others on ROC and MCC metrics.

Table 5 and Fig. 6 show the results of the classifiers based on MAE, MAPE, and RMSE, in which the metrics measure the mean absolute, mean absolute percentage, and root-mean-square errors between the actual and predicted preferences of the customer. The metrics are used to identify how accurate the classifiers are in prediction with lower errors and the deviation of the predicted values from the actual values. The results show that the proposed systems attain lower DP prediction error values than existing schemes. The proposed WOLSTM attain the MAE and RMSE of 1.989 and 2.356.

Table 5 Analysis based on MAE, RMSE and MAPE

Techniques	MAE	RMSE	MAPE
Proposed WOLSTM	1.989	2.356	1.4531
LSTM	2.152	2.624	2.7651
DBN	2.381	3.506	3.2123
CNN	2.786	3.987	4.8761
RNN	3.213	4.678	4.9872

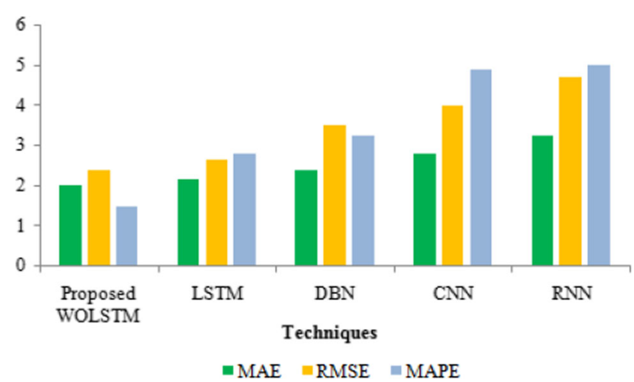

Fig. 6 Analysis based on MAE, MAPE, and RMSE

If the system has a lower MAE, MAPE, and RMSE, then the system is regarded as good. However, the existing LSTM, DBN, CNN, and RNN attain MAE of 2.152, 2.381, 2.786, and 3.123, MAPE of 2.7651, 3.2123, 4.8761, and

4.9872, and RMSE of 2.624, 3.506, 3.987, and 4.678, which are higher than the proposed one. So it is clear that the proposed method attains lower errors for the prediction of DP in EC platforms. These superior results of the proposed method in both detection and error metrics are because it improves the classifier's detection performance with the help of parameter tuning in its network using MSSSO that prevents the network from vanishing gradient and overfitting issues. In addition, the proposed method initially reduces the dimensions of the input data. Then, it groups the customers based on their purchasing behaviour using a clustering approach, leading to the task of DP prediction being more accessible and efficient. The redundant and higher dimension of the selected attributes in the dataset is reduced using the IPCA model, which minimizes the computational burden of the classifier and misleading outcomes in pricing prediction. Using an existing model for testing, such as LSTM, without hyperparameter adjustment produced worse prediction results than the proposed technique. Due to the size of the network, the existing DBN requires a large amount of data to achieve a reasonable level of performance on standard hardware, and its training has proven to be highly costly. Furthermore, overfitting, exploding gradients, and class imbalance were essential hurdles for the CNN during the model's training, and these issues can degrade the model's performance. When using the RNN technique, the training procedure is quite tricky, and the calculation process is prolonged. Because our proposed work addresses these flaws, only the recommended one achieves better results. From the overall experimental analysis, it is concluded that the proposed method achieves superior results than the existing schemes for pricing recommendation of the EC products based on customer's purchasing attributes.

5 Conclusion

Since the popularity of e-commerce has grown significantly during the past decade, it has been the standard for numerous small enterprises to begin by selling their items on such a platform rather than opening a physical store. Given the number of products sold online, dynamic pricing is complicated at this increasing size of online purchasing. The expansion of online markets necessitates the development of a machine-learning tool for pricing recommendations. However, larger ML models require more processing power and are difficult to handle due to the enormous quantity of data in the data set. This research proposes a novel WOLSTM model for the DP recommendation system in EC platforms based on customer purchasing behaviour to address this issue. It mainly consists of five phases: preprocessing, feature extraction, DR, customer grouping, and DP prediction. The proposed system takes the EPPD dataset are taken from Kaggle. The proposed WOLSTM is compared against the LSTM, DBN, CNN, and RNN techniques regarding PN, RL, FE, AY, ROC, MCC, MAE, MAPE, and RMSE metrics. The proposed system achieves a maximum AY of 98.99% with minimum MAE and RMSE of 1.989, 1.4531, and 2.356, respectively. The outcomes of the proposed WOLSTM are superior to the conventional methods. Consequently, the findings show that the best DP prediction strategy considers network externalities to offer price solutions to EC platforms and support the market's steady and orderly growth. A future study might look at meta-heuristic algorithms to improve the accuracy of predicting consumer behaviour. We aim to price products with additional features under more precise conditions. The sales data can also be analysed to determine the other aspects contributing to efficient pricing.

Funding The authors have not disclosed any funding.

Data availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest The authors have not disclosed any conflict of interest.

References

Alghanam OA, Al-Khatib SN, Hiari MO (2022) Data mining model for predicting customer purchase behavior in e-commerce context. Int J Adv Comput Sci Appl 13(2):423–428

Ceci M, Appice A, Loglisci C, Manco G, Masciari E, Ras ZW (eds) (2014) New frontiers in mining complex patterns. Springe, Berlin

Chaubey G, Gavhane PR, Bisen D, Arjaria SK (2022) Customer purchasing behavior prediction using machine learning classification techniques. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-022-03837-6

Chaudhuri N, Gupta G, Vamsi V, Bose I (2021) On the platform but will they buy? Predicting customers' purchase behavior using deep learning. Decis Support Syst 149:113622

Chen H, Xiong W, Xiong P (2022) Pricing analysis of online shopping platforms considering consumer information levels. Front Psychol 13:821979

Chornous G, Horbunova Y (2020) Modeling and forecasting dynamic factors of pricing in e-commerce. In: IT&I, pp 71–82

Dhanalakshmi A, Nagarajan G (2020) Convolutional neural network-based deblocking filter for SHVC in H. 265. Signal Image Video Process 14:1635–1645

Esmeli R, Bader-El-Den M, Abdullahi H (2021) Towards early purchase intention prediction in online session based retailing systems. Electron Mark 31:697–715

Faehnle A, Guidolin M (2021) Dynamic pricing recognition on E-commerce platforms with VAR processes. Forecasting 3(1):166–180

Gao P, Zhao L (2022) Study on deep learning technology to construct e-commerce industry marketing promotion model. Secur Commun Netw 2022:1–11

Guo L (2022) Cross-border e-commerce platform for commodity automatic pricing model based on deep learning. Electron Commer Res 22(1):1–20

Hernes M, Wojtkiewicz K, Szczerbicki E (eds) (2020) Advances in computational collective intelligence: 12th international conference, ICCCI 2020, Da Nang, Vietnam, November 30–December 3, 2020, proceedings, vol 1287. Springer

Kastius A, Schlosser R (2021) Dynamic pricing under competition using reinforcement learning. J Rev Pricing Manag 21:1–14

Leung KH, Luk CC, Choy KL, Lam HY, Lee CK (2019) A B2B flexible pricing decision support system for managing the request for quotation process under e-commerce business environment. Int J Prod Res 57(20):6528–6551

Lou Z, Hou F, Lou X (2021) Optimal dual-channel dynamic pricing of perishable items under different attenuation coefficients of demands. J Syst Sci Syst Eng 30:44–58

Lu CW, Lin GH, Wu TJ, Hu IH, Chang YC (2021) Influencing factors of cross-border e-commerce consumer purchase intention based on wireless network and machine learning. Secur Commun Netw 202(1):1–9

Minu RI, Thyagarajan KK (2013) A novel approach to build image ontology using texton. In: Intelligent informatics: proceedings of the international symposium on intelligent informatics ISI'12 held at august 4–5 2012, Chennai, India. Springer, Berlin, Heidelberg, pp 333-–339

Mohamed MA, El-Henawy IM, Salah A (2022) Price prediction of seasonal items using machine learning and statistical methods. CMC Comput Mater Contin 70(2):3473–3489

Namburu A, Selvaraj P, Varsha M (2022) Product pricing solutions using hybrid machine learning algorithm. Innov Syst Softw Eng. https://doi.org/10.1007/s11334-022-00465-3

Rajalakshmi T, Minu RI (2014) Improving relevance feedback for content based medical image retrieval. In: International conference on information communication and embedded systems (ICICES2014). IEEE, pp 1–5

Safara F (2022) A computational model to predict consumer behaviour during COVID-19 pandemic. Comput Econ 59(4):1525–1538

Simpson SV, Nagarajan G (2021) A fuzzy based co-operative blackmailing attack detection scheme for edge computing nodes in MANET-IOT environment. Future Gener Comput Syst 125:544–563

Tong T, Dai H, Xiao Q, Yan N (2020) Will dynamic pricing outperform? Theoretical analysis and empirical evidence from O2O on-demand food service market. Int J Prod Econ 219:375–385

Victor V, Joy Thoppan J, Jeyakumar Nathan R, Farkas Maria F (2018) Factors influencing consumer behavior and prospective purchase decisions in a dynamic pricing environment—an exploratory factor analysis approach. Soc Sci 7(9):153

Wang Q, Gu B, Fu Y (2022) Optimization of pricing and freshness-keeping effort in fresh e-commerce supply chain considering online comments. Math Probl Eng 2022:1–25

Wu L (2022) Dynamic pricing for dual-channel supply chain considering consumers with double-entry mental accounting. Math Probl Eng 2022:1–13

Xiao M (2022) Supervision strategy analysis on price discrimination of e-commerce company in the context of big data based on four-party evolutionary game. Comput Intell Neurosci 2022:1–25

Xiao L, Guo F, Yu F, Liu S (2019) The effects of online shopping context cues on consumers' purchase intention for cross-border E-commerce sustainability. Sustainability 11(10):2777

Xu J, Wang J, Tian Y, Yan J, Li X, Gao X (2020) SE-stacking: Improving user purchase behavior prediction by information fusion and ensemble learning. PLoS ONE 15(11):e0242629

Zhang P (2021) E-commerce products recognition based on a deep learning architecture: theory and implementation. Future Gener Comput Syst 125:672–676

Zhang W, Wang M (2021) An improved deep forest model for prediction of e-commerce consumers' repurchase behavior. PLoS ONE 16(9):e0255906

Zhao H, Shi Q (2022) Evaluating the impact of community experience on purchase intention in online knowledge community. Front Psychol 13:911594

Zhu L, Lin J (2019) A pricing strategy of E-commerce advertising cooperation in the Stackelberg game model with different market power structure. Algorithms 12(1):24