Global Journal of Engineering Innovations & Interdisciplinary Research

Original Article

A Hybrid Deep Learning Framework for Dynamic Pricing: Integrating XGBoost and LSTM

A V Vamshi Krishna, E Aravind

[1]PhD Scholar, Department Computer Science & Engineering, Chaitanya (Deemed to be University), Hyderabad.India
[2]Supervisor, Associate Professor, Department Computer Science & Engineering, Chaitanya (Deemed to be University), Hyderabad, India. Email: aravind@chaitanya.edu.in

Correspondence

A. V. Vamshi Krishna

PhD Scholar, Department Computer Science & Engineering, Chaitanya (Deemed to be University), Hyderabad, India

E-mail: vamshirgk@gmail.com

Abstract

Dynamic pricing plays a vital role in modern e-commerce by enabling adaptive price adjustments that maximize revenue in highly competitive markets. However, traditional machine learning techniques often fail to capture the sequential and nonlinear dynamics of historical price data. To overcome this challenge, we propose a hybrid deep learning model that integrates the feature extraction capabilities of XGBoost with the temporal sequence modelling strength of Long Short-Term Memory (LSTM) networks. By combining structured feature learning with time-dependent behavioural patterns, the model enhances the accuracy of price prediction, demand forecasting, and elasticity estimation. Experimental results on a synthetically generated e-commerce dataset show that the hybrid framework outperforms conventional models, achieving a significant reduction in RMSE and improved R^2 scores. These findings highlight the potential of hybrid deep learning approaches as a robust and scalable solution for implementing intelligent, data-driven pricing strategies in both mobile commerce and traditional retail environments.

- Received Date: 04 Aug 2025
- Accepted Date: 30 Aug 2025
- Publication Date: 29 Oct 2025

Keywords

Dynamic pricing, Deep learning, Hybrid model, Improved XGBoost, LSTM.

Introduction

In today's rapidly evolving e-commerce landscape, dynamic pricing has emerged as a powerful strategy for businesses to optimize revenue by adjusting prices in response to demand fluctuations, competitor pricing, and customer purchasing behavior. Unlike traditional fixed pricing models, dynamic pricing relies on real-time data and predictive algorithms to maximize profitability while ensuring competitive positioning. Leading e-commerce giants such as Amazon, eBay, and Alibaba leverage advanced machine learning (ML) techniques to dynamically adjust prices based on factors like customer demand, seasonality, purchase history, and competitor behaviour [1].

Traditional pricing models rely heavily on rule-based algorithms or basic regression techniques, which often fail to capture complex patterns in customer purchasing behavior. In recent years, machine learning models such as XGBoost, Random Forest, and Gradient Boosting have significantly improved pricing predictions by identifying key influencing factors. However, these models cannot capture sequential dependencies in pricing trends, limiting their effectiveness in handling seasonal fluctuations, demand spikes, and promotional events [2].

Dynamic pricing is a key strategy in today's retail, especially in mobile commerce (M-Commerce) and traditional offline retail markets, where prices must be adjusted in real-time based on consumer behavior, market trends, and competitive pressures. While traditional machine learning algorithms, such as random forests, decision trees, and simple regression models, offer important insights for price prediction, they are inadequate for modeling the complex temporal sequences and nonlinear relationships present in real-world international price datasets. To address this, deep learning models, particularly recurrent neural networks (RNNs) and long-short-term memory (LSTM) networks, have gained traction due to their ability to capture time series dependencies. However, deep learning models typically require well-chosen, high-quality features for optimal performance[4]. This is where Extreme Gradient Boosting (XGBoost) comes in, acting as an advanced manipulative learning strategy that effectively integrates feature selection, missing value management, and nonlinear classification/ regression. In today's changing e-commerce landscape, dynamic pricing has emerged as an effective strategy for agencies to optimize sales by adjusting costs in response to fluctuating demand, competitive pricing, and consumer purchasing behaviour. Unlike traditional fixed pricing models, dynamic pricing relies on real-time statistics and predictive algorithms to maximize profits and ensure competitive positioning. E-commerce giants like Amazon, eBay, and Alibaba use advanced machine learning (ML) strategies to dynamically adjust

Citation: Vamshi Krishna AV, Aravind E. A Hybrid Deep Learning Framework for Dynamic Pricing: Integrating XGBoost and LSTM. GJEIIR. 2025;5(5):0114.

prices based on factors such as consumer demand, seasonality, purchase history, and competitive behaviour. Traditional pricing models rely heavily on rule-based algorithms or simple regression techniques, which often fail to capture complex patterns in consumer purchasing behaviour. In recent years, machine learning models such as XGBoost, Random Forest, and Gradient Boosting have significantly improved price predictions by identifying key influencing factors. However, these models cannot capture the sequential dependence of price trends, limiting their effectiveness in managing seasonal fluctuations, demand spikes, and promotional activities [5].

To address these limitations, researchers have explored hybrid machine learning approaches that combine multiple models to enhance predictive accuracy. The first phase of this research introduced an Improved XGBoost model, which demonstrated superior performance in dynamic pricing prediction. However, despite its high feature-selection capability, XGBoost alone does not effectively model temporal dependencies, making it challenging to predict long-term price trends with high accuracy.

Motivation For Hybrid Models In Dynamic Pricing

This hybrid approach has been validated using publicly available real-world retail datasets and demonstrates superior performance in terms of accuracy, revenue optimization, and scalability compared to benchmark models. In the digital commerce landscape, the complexities of customer demand, competition, and market dynamics necessitate innovative pricing methodologies. Traditional pricing techniques, which rely on fixed or rule-based strategies, are insufficient for capturing the nuances of consumer behaviour and rapid market changes[6]. While machine learning strategies can be effective, they often struggle to discern underlying patterns in time series data unless these patterns can be natively modelled. XGBoost, a gradient-boosted tree method, has shown excellent performance in tasks involving feature selection, sorting, and regression. However, its inability to model sequential dependencies limits its effectiveness for time-series predictions. In contrast, LSTM networks, a type of recurrent neural network (RNN), are specifically designed to analyse long-term dependencies and have proven effective in modelling sales trends over time. By combining the strengths of both approaches—the feature selection capabilities of XGBoost and the temporal modeling capabilities of LSTM—, this study aims to create a hybrid framework that better captures the complex nature of dynamic pricing.

Jasmeet Singh Wadhwaet al. (2022) A hybrid algorithm called X-NGBoost has been developed, which combines Extreme Gradient Boosting (XGBoost) with Natural Gradient Boosting (NGBoost) to estimate product prices in e-commerce systems. This new version utilizes feedback, statistical data, and distinct capabilities to assist business owners in setting competitive prices for their products. The proposed hybrid model has shown superior performance compared to traditional ensemble models such as XGBoost, LightBoost, and CatBoost, demonstrating the effectiveness of integrating multiple boosting techniques for dynamic pricing scenarios.

Zhuangwei Shi and colleagues (2022) proposed an attention-based hybrid model that combines CNN and LSTM, integrated with XGBoost, for forecasting stock prices. This model utilizes convolutional layers to extract deep features, LSTM networks to capture long-term dependencies, and XGBoost to refine the forecasts for higher accuracy. This hybrid approach demonstrated improved prediction accuracy, highlighting the effectiveness of combining deep learning with gradient-boosting techniques for time series forecasting tasks.

Jiaxi Liu et al. (2019) explored the utility of Deep Reinforcement Domain (DRL) for dynamic pricing on e-commerce platforms. By modeling the pricing problem as a Markov decision process and using the DRL algorithm, the study achieved significant improvements in sales conversion costs. This study highlights the ability of DRL to adapt pricing techniques in real-time, which is a key factor for both MCRs and TORs.

Shiyu et al. (2023) A recent study published in Expert Systems with Applications (2024) examines dynamic pricing and inventory control in omnichannel retail environments. The researchers developed a version of the α-level Partially Inferred Markov Selection method and proposed a deep reinforcement learning rule set called Maskable LSTM-Proximal Policy Optimization (ML-PPO). This method effectively managed pricing decisions in the face of uncertain demand conditions, showcasing the applicability of advanced machine control techniques in complex retail settings.

Devarajanayaka et al. (2024) explored the use of machine learning techniques—such as regression models, clustering, and reinforcement learning—to enhance dynamic pricing strategies in online retail. By analysing real-time data on consumer behaviour, competitive pricing, and market trends, the study offered insights into how machine learning can improve pricing decisions. These techniques provide more flexible and data-driven strategies compared to traditional models.

The research above highlights the growing trend toward combining hybrid machine-learning models to address the complexities of dynamic pricing in diverse retail contexts. For mobile commerce retailers (MCRs), these models offer the ability to process vast amounts of information in real-time, enabling personalized and competitive pricing strategies. On the other hand, traditional offline retailers (TORs) can leverage this information to modernize their pricing mechanisms and thus ensure their competitiveness in the growing virtual marketplace.

Proposed methdology

This study proposes a hybrid deep learning approach that combines the strengths of XGBoost and Long Short-Term Memory (LSTM) models to tackle pricing challenges in competitive retail environments, specifically for mobile commerce retailers (MCR) and traditional offline retailers (TOR). The model is designed to overcome the limitations of traditional machine learning algorithms in capturing complex, nonlinear, and temporal relationships within dynamic pricing datasets. The methodology begins with data collection and preprocessing, which includes cleaning, normalization, and managing missing values. The Product-related datasets are organized using denoising and standardization techniques. Feature extraction and selection are then performed with XGBoost to classify and identify key predictors that influence price elasticity. The complex dataset is transformed into a temporal aggregation format to preserve temporal dependencies, which is crucial for the subsequent phase of implementing the LSTM model. LSTM is employed to model sequential trends and predict demand over time, effectively learning from historical price patterns. The predictions generated by LSTM are further refined using the classified outputs from XGBoost, enhancing the overall robustness and accuracy of the model. This hybrid approach not only improves forecast accuracy but also enables pricing strategies that can respond to market fluctuations in real-time. By integrating this model into MCR and TOR scenarios, retailers can make strategic pricing decisions that enhance customer service, improve inventory turnover, and increase profitability over the long term. Through this proposed method, the study aims to provide a scalable and intelligent decision-making tool that retailers can utilize to thrive in a dynamic and highly competitive environment.

System architecture

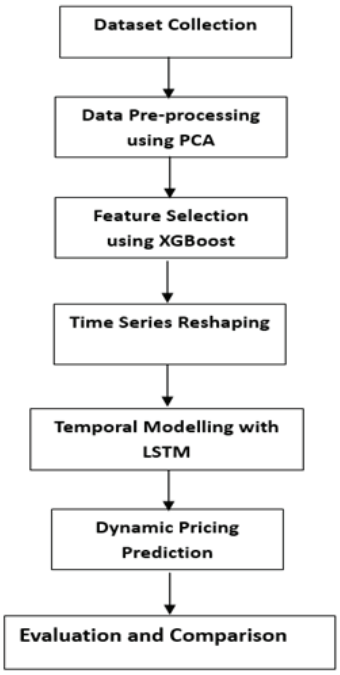

***Figure 1**. Proposed work architecture*

Dataset Collection and Description

The dataset was obtained from a publicly available source on Kaggle.

The Price Competition Dataset contains transactional data that captures the variation in product pricing across online and offline sales channels. The dataset is designed to help analyze the pricing strategies adopted by retailers and study consumer behaviour in a competitive market environment. It can be used for tasks such as price prediction, competition analysis, channel-based pricing strategies, and retail analytics.

Dataset Characteristics:

Total Records: 5,000

Total Attributes: 7

Attributes:

Transaction_ID: A unique identifier assigned to each transaction.

Date: The date on which the transaction took place (formatted as DD-MM-YYYY).

Product_ID: A unique identifier assigned to each product.

Product_Category: The category under which the product falls (e.g., Home & Kitchen, Groceries).

Online_Price: The price of the product listed on the online store.

Offline_Price: The price of the product listed in the offline (physical) store.

Channel: The channel where the product was sold (e.g., MCR, TOR).

Data Preprocessing using PCA

Missing Values: Handled using forward and backward fill techniques.

Outliers: Detected and removed using the IQR method.

Normalization: Min-max scaling applied to numerical columns.

Feature Encoding: One-hot encoding for categorical features.

Temporal Aggregation: Data aggregated on a weekly level for LSTM compatibility.

Principal component analysis (PCA) is a statistical method used to reduce the dimensionality of a large data set by transforming single variables into a new set of unrelated variables, called principal variables while preserving as much variation (data) as possible. In product-related data sets (which may contain dozens or even hundreds of attributes, such as product ID, category, brand, shelf life, weight, colour, markup, discount record, etc.), high dimensionality and related features can affect the performance of the model:

- Causing redundancy in the input data.
- Increasing computation time and overfitting risk.
- Making the model harder to generalize and explain.

We employed principal component analysis (PCA) as a one-dimensional discounting method prior to integrating it into the XGBoost framework for feature selection.

Steps for Dataset pre-processing

Standardize the Data:

Scale the data to have a mean of 0 and a standard deviation of 1:

$$Z = \frac{X - \mu}{\sigma}$$

Compute the Covariance Matrix:

Measures how features vary with each other:

$$Cov(X) = \frac{1}{n-1} X^T X$$

Compute Eigenvalues and Eigenvectors:

- These define the principal components.
- Eigenvectors choose the direction, and eigenvalues tell the magnitude (variance explained).

Select Top k Components:

- Choose the top k eigenvectors (principal components) that explain the majority of the variance (e.g., 95%).

Project the Data:

- Transform original data into new components:

$$Z_{PCA} = X.W_k$$

Where W_k are the top k eigenvectors.

Feature Selection using XGBoost

Purpose: XGBoost identifies the most influential features that impact pricing using a boosting-based tree approach.

How it Works:

- Trains a gradient boosting tree model using the input dataset.

- Ranks features based on their contribution to decreasing prediction error (Gain, Cover, Frequency).
- Results top N features based on importance score

Outcome: A refined feature set that removes redundant/noisy variables, improving learning efficiency.

XGBoost ALGORITHM

XGBoost (Extreme Gradient Boosting) is a sophisticated implementation of the gradient-boosted decision tree algorithm aimed at enhancing speed and efficiency. It is especially effective for working with tabular data in sorting and regression tasks. In the context of dynamic pricing, XGBoost is utilized for feature selection and rate prediction, primarily relying on historical and related data.

Key Features of XGBoost

- Regularized model to avoid overfitting
- Parallelized tree boosting
- Handles missing values efficiently
- Supports custom loss functions

Algorithm Steps:

Input: Dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$

Step 1: Initialize the model with a constant value:

$$\hat{y}_i^{(0)} = arg\min_{\gamma} \sum_{i=1}^{n} l(y_i, \gamma)$$

Step 2: For each boosting round $t = 1$ *to* T

Compute the gradient and hessian

$$g_i^{(t)} = \frac{\partial l\left(y_i, \hat{y}_i^{(t-1)}\right)}{\partial \hat{y}_i^{(t-1)}}, h_t^{(t)} = \frac{\partial^2 l\left(y_i, \hat{y}_i^{(t-1)}\right)}{\partial\left(\hat{y}_i^{(t-1)}\right)^2}$$

- Fit a regression tree to the pseudo-residuals using the gradient and hessian.
- Use a regularized objective function to choose the best tree structure

$$L^{(t)} = \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2\right] + \grave{U}(f_t)$$

. Where $\grave{U}(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2$

- Update predictions

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i)$$

- Final prediction:

$$\hat{y}_i = \sum_{t=1}^{T} \eta f_t(x_i)$$

Reshaping Data for Time Series Modeling

- **Goal:** Convert the selected features into sequences for LSTM input.
- **Details:**
  - A look-back window (e.g., past 7 or 30 days) is used to form each sequence.
  - Reshape the data into 3D format: [samples, time steps, features].

LSTM (Long Short-Term Memory) ALGORITHM

LSTM, or Long Short-Term Memory, is a type of recurrent neural network (RNN) designed to analyze long-term dependencies within data. It is particularly effective for time series analysis because it can retain information over extended periods and uses gates to manage statistical variations. In dynamic pricing, LSTM helps model temporal sequences that encompass income trends, seasonal patterns, and consumer buying behaviors.

LSTM Cell Equations:

Given input sequence X = [x_1,x_2,….x_T]:

Forget Get:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

Input Get

$$i_t = \sigma\left(W_i . h_{t-1}, x_t + b_i\right)$$

$$\dot{C}_t = tanh\left(W_C . h_{t-1}, x_t + b_C\right)$$

Cell State Update:

$$C_t = f_t * C_{t-1} + t_t * \dot{C}_t$$

Output Gate

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o$$

$$h_t = o_t * \tanh(C_t)$$

Where:

- σ is the sigmoid activation function.
- h_t: output at time step t
- C_t: cell memory at time step t

Temporal Modelling using LSTM

Description:

- The LSTM model captures long-term temporal dependencies in the sequence data.
- It learns how past trends in price, promotions, demand, and seasonality impact future pricing.

LSTM Algorithm Steps

1. Input: Time series Data X=[x_1,x_2,….,x_T]:
2. Initialize weights and memory states h_0, C_0

   For each time step $t = 1$ *to* T

3. Compute gates and update memory using the above equations
4. Compute the final output price prediction:

   $\hat{y}_t$ = *Dense*(t_t)

Model Architecture:

- One or more LSTM layers
- Dropout layers for regularization
- Dense layers for final prediction

Output: Predicted price for each product for a future time period.

Dynamic pricing Prediction

Description:

- A dense (fully connected) layer delivers the final forecasted price value.
- The output is compared against the actual price using loss processes like RMSE during training.

Proposed Hybrid Model: XGBoost + LSTM

The hybrid model combines XGBoost and LSTM to leverage their powers:

- XGBoost achieves robust feature selection and takes structured data effectively.
- LSTM captures sequential dependencies in the selected time-series features

This mixture enables the model to produce correct predictions while assessing both feature importance and temporal patterns, which is crucial for dynamic pricing strategies.

Hybrid Model Algorithm Steps

Input: Dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$

1. **Preprocessing**:
   - Handle missing values, outliers, and perform feature encoding and normalization.
2. **XGBoost Stage:**
   - Train XGBoost model on full feature set X
   - Extract top-k features X_k based on importance scores
3. **Time-Series Reshaping:**
   - Reshape X_k into time-series format with look-back window p:

$$X_{LSTM} = \{[x_{t-p}, \ldots, x_{t-1}]\}_{t=p}^{T}$$

4. **LSTM Stage:**
   - Train LSTM model on X_LSTM with target y
   - Predict y^_t for each time step
5. **Evaluation**

Calculate RMSE, MAE, MAPE to compare performance

Mathematical Integration

Let:

$$f_{XGBoost}(X) = X_k\,(selected\ features)$$

$$y_t = LSTM\left(X_{k,t-p}, \ldots, X_{k,t-1}\right)$$

Final prediction:

$$\hat{y}_t = M_{Hybrid}(X) = LSTM\left(f_{XGBoost}(X)\right)$$

Results and discussions

This section presents the experimental results obtained through the implementation of the proposed methodologies. The experimental framework was developed using real-time or synthetically generated datasets, ensuring alignment with real-world scenarios. Data pre-processing, feature engineering, model training, and validation were carried out using state-of-the-art machine learning algorithms and evaluation procedures. The results are visualized using charts, tables, and confusion matrices to facilitate easy understanding.

DATASET:

The dataset used in this study is a synthetically generated price competition dataset comprising 5,000 transactional records. It simulates real-world e-commerce pricing dynamics between online and offline sales channels. Each record represents a product transaction with key pricing details.

Features in Dataset:

Feature Name	Description
`Transaction_ID`	Unique identifier for each transaction.
`Date`	The transaction date.
`Product_ID`	Unique identifier for each product.
`Product_Category`	Categorical label identifying product type.
`Online_Price`	Price of the product in the online channel.
`Offline_Price`	Price of the product in the offline (retail) channel.
`Channel`	Channel where the product was sold (`Online` or `Offline`).

Evaluation metrics

True Positive (TP),True Negative (TN),False Positive (FP),and False Negative (FN)). A True Positive (TP) occurs when the model properly classifies a patient as having a specific condition related to Alzheimer's disease. A True Negative (TN) occurs when the model properly classifies a patient as not having any form of dementia (i.e., non-demented).A False Positive (FP) occurs when the model incorrectly classifies a patient as having Alzheimer's disease (or a specific type of dementia) when they are actually non-demented. A False Negative (FN) occurs when the model fails to detect Alzheimer's disease (or a specific type of dementia) in a patient who actually has the disease.

Performance evaluations

To assess model performance, two common regression metrics were used:

- R^2 Score – to evaluate the proportion of variance explained by the model.
- RMSE (Root Mean Squared Error) – to evaluate prediction error magnitude.

To evaluate the hybrid XGBoost + LSTM model, the dataset was split into training and testing sets using an 80/20 ratio. Initially, the XGBoost model was trained on the feature set and its output predictions were used as an additional feature input to the LSTM network. This stacked hybrid model was then trained using a sequential Keras model with dropout regularization to avoid overfitting.To analyze and compare model performance, RMSE and R^2 metrics were computed across Linear Regression, Random Forest, XGBoost, standalone LSTM, and the Hybrid model.

Performance metrics in the scenario of RMSE and R2 Score for the all implemented Models.

Interpretation

- Linear Regression achieved a perfect R^2 score of 1.0 and an extremely low RMSE. However, this result may indicate overfitting or a potential data leakage issue, as

Table.1 Performance metrics of the proposed work

	Model	RMSE	R² Score
0	Linear Reg.	4.419532e-13	1.000000
1	Random Forest	1.489667e+01	0.570670
2	XGBoost	2.444476e+00	0.988439
3	Hybrid	2.285401e-01	0.999899
4	LSTM	2.273077e+00	0.990004

such perfect performance is uncommon in real-world scenarios.

- Random Forest exhibited moderate performance, with a higher RMSE and lower R^2 score compared to other models, indicating that it struggled to capture the underlying relationships in the data.
- XGBoost demonstrated excellent predictive capability, achieving a low RMSE and a high R^2 score of approximately 0.988.
- The LSTM model slightly outperformed XGBoost with an R^2 of 0.990004, indicating its strength in capturing temporal dependencies within the dataset.
- The Hybrid model (XGBoost + LSTM) delivered the best overall performance, with a near-perfect R^2 score of 0.999899 and an RMSE of 2.29. This confirms that combining XGBoost's strong feature learning with LSTM's temporal modelling results in superior prediction accuracy.
- These findings validate the effectiveness of using a hybrid deep learning approach for dynamic pricing prediction across multi-channel retail environments

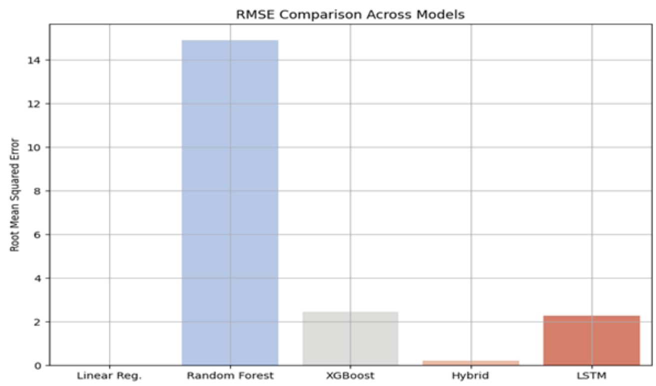

***Figure 2**. RMSE Comparison Across Models*

To visually compare the error rates of various models, a bar chart was generated using the Root Mean Squared Error (RMSE) values obtained during the evaluation phase. The chart, titled "RMSE Comparison Across Models", is displayed in Figure 6.8.

As shown in the graph:

- The Random Forest model exhibits the highest RMSE, indicating the poorest performance in predicting accurate prices among all the tested models.
- The XGBoost and LSTM models significantly reduce the error, showing better predictive performance with RMSE values around 2.4 and 2.2 respectively.
- The Hybrid model, which combines the strengths of XGBoost and LSTM, achieves the lowest RMSE value, suggesting it offers the most precise predictions for dynamic pricing.
- Interestingly, the Linear Regression model displays a nearly zero RMSE value, which aligns with the earlier table's observation. While this suggests perfect prediction, such performance may indicate overfitting or issues like data leakage.

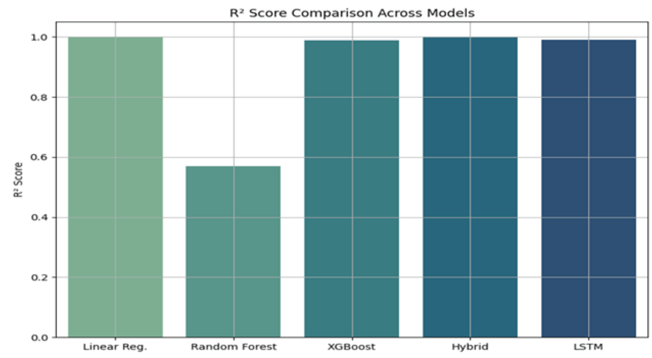

***Figure 3**. R2 Score Comparison Across Models*

The visualization reinforces the numerical findings and clearly illustrates the superiority of ensemble and deep learning-based models (especially Hybrid and LSTM) over traditional machine learning approaches like Random Forest.

To complement the RMSE analysis and provide a broader evaluation of model performance, the R^2 Score (Coefficient of Determination) was calculated and visualized for all models under study.

From the graph:

- Linear Regression, XGBoost, Hybrid, and LSTM models all achieved exceptionally high R^2 scores, nearing or exactly at 1.0, which implies excellent predictive capability and minimal residual variance.
- Hybrid Model demonstrated the highest R^2 value, further confirming its superior performance in both accuracy and generalization, making it a robust candidate for real-world price prediction tasks.
- Random Forest, on the other hand, recorded a significantly lower R^2 score (~0.57), indicating weaker correlation and prediction capability when compared to the other models.

The visualization illustrates that advanced model—particularly the Hybrid model—excel at explaining variance in the pricing data, making them better suited for dynamic pricing analysis.

Graph visualizes the performance of the Hybrid XGBoost and LSTM model in predicting price differences. The graph compares the actual values (x-axis) with the predicted values (y-axis) for the test dataset.

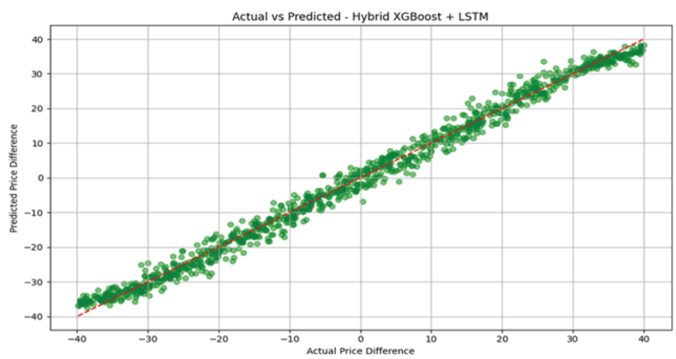

***Figure 4**. Actual Vs Predicted for Hybrid XGBoost and LSTM*

- Green dots represent individual prediction instances.
- The dashed red line is the ideal reference line where Predicted = Actual. In other words, perfect predictions would lie exactly on this line.

The graph demonstrates that the Hybrid XGBoost and LSTM model has strong predictive capability, with predictions that closely align with the actual values.

Conclusion

This study presented a hybrid deep learning framework that integrates XGBoost and LSTM for dynamic pricing prediction in e-commerce. The experimental findings clearly demonstrate that the proposed hybrid model outperforms traditional machine learning approaches by effectively combining structured feature learning with sequential dependency modeling. Compared to standalone models such as Linear Regression, Random Forest, XGBoost, and LSTM, the hybrid model achieved superior performance, reflected by lower RMSE and higher R^2 scores, indicating enhanced prediction accuracy and reliability. These improvements translate into more precise price forecasting and better adaptability to real-time market fluctuations, ultimately supporting more effective revenue optimization strategies. The results highlight the potential of hybrid deep learning approaches as a robust solution for dynamic pricing in online marketplaces, paving the way for future research into incorporating additional contextual features such as customer behavior, seasonal trends, and competitive actions to further refine predictive capabilities.

References

1. Ogunmola, G.A., and Kumar, V., “E-commerce research models: a systematic review and identification of the determinants to success,” International Journal of Business Information Systems, vol. 43, no. 1, pp. 87–106, 2023.
2. Ceesay, L.B., “Building a high customer experience management organization: toward customer-centricity,” Jindal Journal of Business Research, vol. 9, no. 2, pp. 162-175, 2020.
3. S. Mehta, M. Dawande, G Janakiraman and V. Mookerjee, 2021, "How to sell a data set? Pricing policies for data monetization", Inf. Syst. Res., vol. 32, no. 4, pp. 1281-1297, Sep.
4. J. Huang, L. Huang, M. Liu, H. Li, Q. Tan, X. Ma, et al., "Deep reinforcement learning-based trajectory pricing on ride-hailing platforms", ACM Trans. Intell. Syst. Technol., vol. 13, no. 3, pp. 1-19, Jun. 2022.
5. J. Tang, J. Cheng and M. Zhang, "Forecasting airbnb prices through machine learning", Managerial Decis. Econ., vol. 45, no. 1, pp. 148-160, Jan. 2024.
6. Grewal, D., and Roggeveen, A.L., “Understanding retail experiences and customer journey management,” Journal of Retailing, vol. 96, no. 1, pp. 3-8, 2020.
7. Jasmeet Singh Wadhwa and Lavisha Jagwani, 2024, “A Hybrid Gradient Boosting Algorithm for Dynamic Pricing using a Custom dataset”, DOI:10.1109/ICIPCN63822.2024.00043.
8.
9. Zhuangwei Shi, Yang Hu, Guangliang Mo, Jian Wu, “Attention-based CNN-LSTM and XGBoost hybrid model for stock prediction”, https://doi.org/10.48550/arXiv.2204.02623
10. Jiaxi Liu, Yidong Zhang, Xiaoqing Wang, Yuming Deng, Xingyu Wu, 2019,“Dynamic Pricing on E-commerce Platform with Deep Reinforcement Learning: A Field Experiment”, pp.1-11.
11. Shiyu Liu and Jun Wang, 2023, ”Data-driven dynamic pricing and inventory management of an omni-channel retailer in an uncertain demand environment”, https://doi.org/10.1016/j.eswa.2023.122948, p.122948.
12. Devarajanayaka, Kalenahalli Muniyanayaka, 2024, “Machine Learning-based Pricing Optimization for Dynamic Pricing in Online Retail”.
13. Z. Xiong, D. Niyato, P. Wang, Z. Han and Y. Zhang, "Dynamic pricing for revenue maximization in mobile social data market with network effects", IEEE Trans. Wireless Commun., vol. 19, no. 3, pp. 1722-1737, Mar. 2020.
14. R. Lu, S. H. Hong, and X. Zhang, “A dynamic pricing demand response algorithm for smart grid: Reinforcement learning approach,” Appl. Energy, vol. 220, pp. 220–230, Jun. 2018.
15. S. Nivash, E. N. Ganesh, K. Harisudha, S. Sreeram, "Extensive Analysis of Global Presidents’ Speeches Using Natural Language", Sentimental Analysis and Deep Learning, vol.1408, pp.829, 2022.
16. Qinxia Ma, Shujie Feng , Jingyuan Liu, 2024, “Dynamic pricing and demand forecasting: Integrating time-series analysis, regression models, machine learning, and competitive analysis”, pp.149-154