Explainable LSTM-Based Demand Forecasting for Multi-Channel Retail Supply Chain Performance Analysis

Tran Ha Thu and Quang-Anh N.D., Hai Yen Nguyen, Thao Phuong Pham, Duong Le Tung Ba, Chi Tuong La, Anh Minh Hoang, Thi-Minh-Ngoc Luu, Zhang Yuemei, Ha Manh Hung*

International School, Vietnam National University, Xuan Thuy, Cau Giay, Hanoi, Vietnam

*Corresponding author: hunghm@vnu.edu.vn

Abstract

Modern retail supply chains face unprecedented complexity due to multi-channel distribution, demand volatility, and increasing customer expectations. This study presents a comprehensive evaluation framework integrating ten state-of-the-art machine learning algorithms with explainable AI methodologies to address both predictive accuracy and model transparency in supply chain demand forecasting. We evaluate diverse algorithmic families including ensemble methods, gradient boosting variants, kernel machines, and recurrent neural networks on the DataCo Global multi-channel retail supply chain dataset (2016-2018, 180,519 transactions). Experimental results reveal LSTM's exceptional performance achieving RMSE of 42.20, R^2 of 98.79%, and MAPE of 0.41%, demonstrating 39-fold improvement over traditional approaches. The integrated explainable AI framework using LIME unveils temporal lag features and delivery scheduling as critical demand determinants, transforming black-box predictions into actionable business intelligence for supply chain practitioners.

Research purpose:

To develop an evaluation framework bridging predictive performance and model interpretability in retail supply chain demand forecasting through systematic comparison of advanced machine learning approaches.

Research motivation:

Contemporary supply chain forecasting research lacks comprehensive frameworks jointly optimizing prediction accuracy and model transparency. Multi-channel retail operations require forecasting systems delivering both superior performance and explainable insights for strategic inventory management, distribution planning, and operational decision-making.

Research design, approach, and method:

Multi-algorithmic evaluation of ten algorithms on real-world supply chain transaction data spanning two years. Models evaluated using RMSE, R^2, and MAPE metrics, with LIME-based interpretability analysis to identify key demand drivers and validate model reasoning.

Main findings:

LSTM achieves superior temporal dependency modeling with 98.79% R^2 accuracy. Gradient boosting methods show competitive performance while ensemble approaches reveal metric inconsistencies suggesting overfitting. Explainability analysis identifies temporal lag features, moving averages, and seasonal patterns as primary predictive drivers, with delivery timing significantly influencing demand patterns.

Practical/managerial implications:

The framework enables supply chains to transition from reactive to predictive management paradigms. LSTM-driven forecasting optimizes inventory positioning, reduces stockouts, and minimizes holding costs. Explainable components build stakeholder trust and provide actionable insights for operational adjustments.

Keywords: Sales Forecasting, Supply Chain Management, LSTM, Explainable AI, Deep Learning, Retail Operations

N. D. Nguyen and P. T. K. Ngoc (eds.), *Proceedings of the International Conference on Emerging Challenges: Business Dynamics in Disruptive Economy (ICECH 2025)*, Advances in Economics, Business and Management Research 377,
https://doi.org/10.2991/978-94-6239-622-7_33

1. INTRODUCTION

Multi-channel retailing spanning online platforms, physical stores, and hybrid fulfillment models has intensified supply-chain planning complexity According to Wu, S., et al. (2024). Demand now reflects overlapping promotional cadences, short fulfillment cycles, cross-channel cannibalization and halo effects, and calendar- or event-driven surges; in such settings, forecast errors rapidly translate into stock-outs, excess safety stock, expedited shipping, and margin erosion. Compared with traditional channels, e-commerce adds further complications: promotion- and event-sensitive spikes (flash sales, markdowns, free-shipping campaigns, shopping festivals), short product life cycles and frequently changing assortments that create cold-start and intermittency, heightened service-level expectations with short lead times, and rich but volatile digital traffic signals (impressions, sessions, click-through, add-to-cart) that interact with weekly and seasonal patterns. These characteristics elevate the value of sequence models capable of learning temporal dependencies and regime shifts. Over the past few years, the rapid growth of online shopping has intensified the demand for efficient and resilient supply chain systems. Global e-commerce sales reached USD 6.33 trillion in 2023 and are projected to rise to USD 8.03 trillion by 2027, introduced by Torres, D., et al. (2024)., underscoring the urgency for businesses to enhance their supply chain capabilities. The term e-commerce describes transactions for goods and services carried out over computer networks through platforms designed for ordering, and it now stands as one of the most dynamic parts of world trade, studies by OECD et al. (2019). Managing supply chains in such an environment not only requires productivity, but the understanding of consumer demand, optimizing the inventory, and precise and timely delivery. Demand forecasting is particularly critical in e-commerce because inaccurate forecasts can lead to stockouts, excess inventory, delivery delays, and dissatisfied customers, which are magnified in highly competitive digital marketplaces, introduced by Li, J., et el. (2021). In this regard, predictive analytics has become indispensable in contemporary supply chain management as it moves businesses from reactive to proactive decision-making.

Machine learning (ML) has significantly transformed the way demand forecasting and supply chain optimization are approached by Jahin, M. A., et al. (2025). While traditional models such as Linear Regression continue to be widely applied, they are now complemented by more advanced algorithms. Algorithms like Decision Tree, LightGBM, XGBoost, Random Forest and Gradient Boosting provide extremely useful tools for handling complex, high-dimensional data. Similarly, methods such as Support Vector Machines (SVM), Multilayer Perceptron (MLP), AdaBoost, and Long Short-Term Memory (LSTM) networks have been found to be effective predictors of nonlinear relationships and dynamic buyer behavior. Beyond ML, the growing field of Deep Learning (DL) offers powerful architectures capable of capturing long-term dependencies, nonlinear dynamics, and unstructured data, thereby expanding the predictive toolkit for supply chain analytics, introduced by Kim, J., et al. (2025). Such methods allow firms to make more accurate forecasts, reduce inventory risks, and lower operating costs. Aside from prediction, the explainability of machine learning models also becomes more critical in supply chain research and practice, approached by Moktadir, M. A., et al. (2019). Methods like LIME (Local Interpretable Model-Agnostic Explanations) introduce transparency by identifying the most important factors affecting model predictions. This not only enhances dependence on predictive analytics but also provides actionable advice for enhancing decision-making frameworks by managers.

Despite their potential, the application of advanced ML, DL, and Explainable AI (xAI) in multi-channel retail supply chain, particularly within fast-growing e-commerce markets, remains underexplored by Zhang, Q. (2025). A recent review of 119 papers found that although many studies apply ML and DL for demand forecasting, relatively few integrate interpretability methods introduced by Douaioui, K., et al. (2024). Moreover, other recent works show ML/AI methods reducing forecasting error by about 10-20% under disruption conditions, introduced by Balan, G. S., et al. (2025), and adaptive routers improving accuracy by ~11.8% over strong benchmarks with faster inference times by Zhang, Q. (2025). However, research that systematically evaluates both the predictive power and interpretability of ML and DL models in supply chain contexts is still underdeveloped by Balan, G. S., et al. (2025). For example, while Mitchell *et al.* (2019) provided an overview of ML adoption in supply chains, more recent work such as Jahin *et al.* (2025) demonstrates that even cutting-edge explainable deep learning frameworks still encounter difficulties in achieving an optimal trade-off between model accuracy and interpretability.

This study seeks to bridge this gap by introducing and evaluating an integrated framework that merges machine learning forecasting approaches with the explainability offered by LIME. This goal is to enhance the accuracy of demand forecasting, increase the efficiency of logistics operations, and enhance the overall effectiveness of the supply system, while providing interpretable insights that support managerial decision processes, furthermore the overview of our proposed architecture for multi-channel retail supply chain forecasting is outlined as in Figure 1.

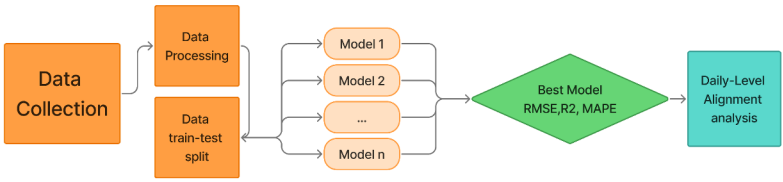

Fig. 1. Overall architecture of our processing pipeline

2. LITERATURE REVIEW

2.1 AI in Supply Chain Forecasting

In supply chain forecasting, ML has transformed traditional approaches by capturing nonlinear relationships, handling high-dimensional data, and integrating auxiliary variables such as promotions, prices, and macroeconomic indicators. Tree-based methods are valued for their interpretability and robustness on tabular data, while boosting algorithms such as XGBoost represent a major advancement by combining weak learners into a strong learner, integrating regularization, and optimizing directly on real-world loss functions as introduced by Chen, T., & Guestrin, C. (2016, August). One of the appealing characteristics of tree-based and boosting methods is that categorical features, missing values, and table inputs with high-dimensional settings are dealt with little preprocessing and yet often outperform deep neural networks with structured data settings.

However, e-commerce supply chains are characterized by complex seasonal patterns, abrupt shocks (*e.g.*, flash sales), and long-term dependencies, where DL architectures such as LSTM networks or transformers show superior performance in capturing sequential dependencies. Empirical evidence suggests that DL models excel when the dataset is large and time-series signals are strong, whereas boosting methods often dominate on high-dimensional tabular data with multiple contextual variables, especially in resource-constrained environments due to faster training and inference Chen, T., & Guestrin, C. (2016, August). Consequently, model choice in supply chain forecasting requires balancing the nature of data (tabular vs sequential), dataset scale, and deployment objectives (batch forecasting vs real-time prediction).

2.2 Explainable AI in Forecasting

As predictive analytics becomes central to supply chain decision-making, model interpretability has emerged as a critical requirement for managerial trust and adoption. Two major families of xAI methods are local explanations and global explanations. LIME approximates a model's decision boundary around a single prediction with a simple surrogate model, highlighting the most influential features locally.

Applied to supply chain forecasting, xAI enables managers to identify demand drivers (*e.g.*, promotions, seasonality, or price shocks), validate model outputs through sanity checks, and detect bias in data prior to deployment. Nevertheless, two challenges limit real-world deployments: (i) stability and reliability with LIME explanations can vary depending on sampling strategies and surrogate parameters; and (ii) computational cost.

2.3. Strengths and Limitations of Previous Studies

Existing studies provide strong evidence that boosting algorithms, particularly XGBoost, significantly reduce forecasting errors in retail and logistics compared to regression-based methods such as Chen, T., & Guestrin, C. (2016, August). Meanwhile, comprehensive reviews of interpretability frameworks emphasize the trade-offs between accuracy, stability, and computational efficiency of LIME identifying their potential but also their fragility in practical deployment. In addition, technical efforts such as GPU accelerated implementations have made boosting algorithms more scalable and suitable for near real-time applications, Mitchell, R., et el. (2018).

Despite these advances, the literature reveals notable limitations. First, most studies emphasize statistical accuracy metrics (*e.g.*, RMSE, MAPE) while neglecting managerial usability whether explanations meaningfully improve operational decision-making. Second, even though LIME provides mathematically rigorous feature attributions, its outputs can be difficult for non-technical managers to interpret in highly interactive, time dependent data. This suggests a need for explanation methods that combine numerical attributions with intuitive visualization and human in the loop validation. Third, many studies benchmark single models in static environments, with limited comparative analysis of boosting versus deep learning under disruption scenarios such as pandemic-related demand shocks, flash sales, or supply interruptions. This restricts the generalizability of existing findings as Mitchell, R., et el. (2018).

In summary, while the foundations of boosting and XAI are mature and effective for building high-performance forecasting systems, the literature still lacks systematic benchmarking across (i) data modalities (tabular vs sequential), (ii) computational trade off when embedding XAI in real-time pipelines, and (iii) resilience under highly volatile e-commerce conditions. These unresolved gaps provide the motivation for the present study, which proposes and evaluates an integrated framework combining advanced ML/DL models with LIME to jointly address accuracy, interpretability, and robustness.

3. METHODOLOGY

3.1. Model Selection

In this research, we utilize a broad range of ML and DL techniques to predict sales in the supply chain. The models considered encompass tree-based methods, ensemble learning approaches, kernel-driven algorithms, gradient boosting techniques, and neural networks. This selection is intended to deliver a thorough evaluation of how effectively these algorithms can address complex and nonlinear supply chain data.

3.1.1. Long Short-Term Memory

LSTM networks are a variant of RNNs built to handle long-term dependency issues in time series shown by Hochreiter, S., & Schmidhuber, J. (1997). Unlike traditional RNNs, LSTM incorporates hidden units with a memory cell and gating mechanisms, enabling the network to retain important information over extended time steps while filtering out irrelevant information. This mechanism allows LSTM to learn and predict complex and nonlinear relationships in time series that traditional RNNs struggle with, particularly when critical events are temporally distant. Each LSTM unit is composed of a cell state and three main gates: the input gate i_t, the forget gate f_t, and the output gate o_t. The internal computations of an LSTM cell can be expressed through the following set of equations:

$$f_t = \sigma(Q_t \times [h_{t-1}, x_t] + b_t) \tag{1}$$

$$i_t = \sigma(Q_i \times [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\underline{C_t} = tanh(Q_C \times [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \underline{C_t} \tag{4}$$

$$o_t = \sigma(Q_o \times [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \odot tanh(Q_t) \tag{6}$$

Here x_t is the input at time step t, h_{t-1} is the hidden state from the previous step, and $C_{t-1}$is the previous cell state, σ denotes the sigmoid function, $\odot$ represents the Hadamard product, and Q and b are the learnable weights and biases.

3.1.2. Extreme Gradient Boosting (XGBoost)

XGBoost is an algorithm within the Gradient Boosting Machines (GBM) family, renowned for its high performance in supervised learning tasks. It can be applied to both regression and classification problems, one of the reasons XGBoost is widely favored is its fast computational speed and its ability to handle large datasets through out-of-core computation

Mitchell, R., et el. (2018). The functioning of XGBoost relies on the boosting technique, where multiple weak decision trees are sequentially combined, with each new tree built to minimize the errors made by the preceding ones, thus enhancing both the model's prediction ability and training accuracy. The algorithm achieves this by optimizing an objective function that incorporates a loss component along with a regularization term, formulated as:

$$Obj = \sum_{i=1}^{n} \quad l(y_i \hat{y}_i) + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \quad w_j^2 \tag{7}$$

Here, T represents the total number of leaves, w_j denotes the weight assigned to leaf j, and γ, λ are regularization parameters used to control model complexity and mitigate overfitting. To enable efficient optimization, XGBoost employs a second-order Taylor expansion of the loss function, as shown in Equation (8).

$$bj^{(t)} \approx \sum_{i=1}^{n} \quad [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \quad w_j^2 \tag{8}$$

where g_i *and* h_i represent the first and second order gradients of the loss function with respect to the prediction obtained from the (t−1)−th iteration. The optimal weight assigned to each leaf can then be computed as follows:

$$w_j^* = \frac{\sum_{i \in I_j} \quad g_i}{\sum_{i \in I_j} \quad h_i + \lambda} \tag{9}$$

with I_j representing the set of training samples assigned to leaf j.

3.1.3. Random Forest

Random Forest (RF) is a powerful ensemble learning algorithm that utilizes a collection of decision trees and can be applied to both regression and classification tasks introduced by Salman, H. A., et el. (2024). The method generates multiple decision trees using bootstrap samples of the training data. This ensemble strategy reduces overfitting and improves predictive performance compared to a single tree. The model's behaviour primarily depends on two hyperparameters, the number of trees and the number of randomly selected features at each split. For regression problems, the overall prediction corresponds to the mean value of all tree outputs:

$$\hat{y} = \frac{1}{B} \sum_{b=1}^{B} \quad T_b(x) \tag{10}$$

where B is the number of trees, and $T_b(x)$ is the prediction of the b^{th} decision tree for input x, for classification, RF uses majority voting as outlined in Equation (11).

$$\hat{y} = mode\{T_1(x), T_2(x), \ldots, T_B(x)\} \tag{11}$$

3.1.4. Gradient Boosting

Gradient Boosting (GB) is one of the ensemble based techniques applied in both regression and classification tasks as Friedman, J. (2001). It constructs a series of weak learners - commonly decision trees, where each successive tree aims to reduce the mistakes made by earlier ones. The algorithm improves performance by repeatedly minimizing a differentiable loss function through gradient descent. GB is flexible, capable of modelling complex nonlinear relationships and feature interactions. Proper regularization techniques, such as shrinkage, subsampling, and tree depth limitation, are

used to prevent overfitting. The objective function is defined by $Obj = \sum_{i=1}^{n} \quad (y_i \hat{y}_i)$, at iteration t, the model is updated by:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i) \tag{12}$$

where η is the learning rate, and $f_t(x)$ is the weak learner fitted to the pseudo-residuals:

$$r_{it} = -[\frac{\partial l(y_i \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}] \tag{13}$$

3.1.5. Decision Tree

Decision Tree (DT) is a widely used ML algorithm, developed from early research on decision tree models in the 1960s, with significant contributions by Ross Quinlan through the ID3 and C4.5 algorithms, Quinlan, J. R. (1986). DT can be utilized for regression and classification tasks, relying on a hierarchical tree-like structure. In DT, internal nodes denote tests on features, branches indicate the results of those tests, and leaf nodes provide the final prediction, either as a class label or a numerical value. The tree is built by recursively splitting the dataset from the root down to the leaves, using measures such as Entropy, Gini Index, or Mean Squared Error (MSE), until certain stopping rules are satisfied, for example, reaching the maximum depth or the minimum number of samples in a leaf. One of the main strengths of DT lies in its transparency, as it allows users to follow the decision-making process of DT from the root node to the output leaf.

3.1.6. CatBoost

Proposed by Yandex in 2017 CatBoost is a gradient boosting algorithm notable for its ability to efficiently handle categorical features and reduce overfitting on real-world data, Dorogush et el. (2018, October). Unlike traditional gradient boosting methods, CatBoost employs an ordered boosting scheme and advanced categorical feature encoding, enhancing generalization capabilities while maintaining high accuracy on complex datasets. The algorithm sequentially constructs an ensemble of decision trees, with each new tree trained to minimize the residual errors of the previous trees. CatBoost utilizes symmetric trees with balanced structures across branches, which accelerates training and simplifies inference. Techniques such as ordered target statistics and random permutations prevent target leakage and ensure unbiased gradient estimation, making the algorithm suitable for small, medium, and complex datasets with high-cardinality categorical variables.

3.1.7. Support Vector Machine

Proposed by Vapnik, Support Vector Machine (SVM) is a supervised learning method that can be employed for both classification and regression tasks in Schölkopf, B., et al. (2000). Its core goal is to determine the optimal hyperplane in the feature space that best separates data classes while maximizing the margin - the distance between this hyperplane and the closest data points from each class. A distinctive feature of SVM is the application of structural risk minimization, which facilitates the derivation of a decision function with strong generalization performance.

3.1.8. Light gradient-Boosting Machine

Similar to XGBoost, Light Gradient-Boosting Machine (LightGBM) supports parallel computation of decision trees, but it stands out for faster training speed and more efficient memory usage, Ke, G., et el. (2017). LightGBM's efficiency derives from three main components: (1) the Gradient-based One-Side Sampling (GOSS) algorithm, which preserves high-gradient samples while subsampling low-gradient ones to reduce computational overhead; (2) the Exclusive Feature Bundling (EFB) technique, which groups mutually exclusive features into bundles for faster processing; and (3) a leaf-wise growth strategy with histogram-based optimization and depth limitation, allowing the model to achieve high accuracy while controlling overfitting. The objective function is defined as:

$$bj = \sum_{i=1}^{n} \quad l(y_i \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{14}$$

Using second-order Taylor expansion:

$$Obj^{(t)} \approx \sum_{i=1}^{n} \quad (g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 + \Omega(f_t) \tag{15}$$

Tree growth is guided by the maximum gain criterion:

$$Gain = \frac{1}{2}\left(\frac{(\Sigma_{i\in L} g_i)^2}{\Sigma_{i\in L} h_i + \lambda} + \frac{(\Sigma_{i\in R} g_i)^2}{\Sigma_{i\in R} h_i + \lambda} - \frac{(\Sigma_{i\in (L\cup R)} g_i)^2}{\Sigma_{i\in (L\cup R)} h_i + \lambda}\right) - \gamma \tag{16}$$

3.1.9. Adaptive Boosting

Adaptive Boosting (AdaBoost) is an ensemble learning based algorithm introduced by Freund and Schapire in 1996, Freund, Y., & Schapire, R. E. (1996, July). The core idea of AdaBoost is to sequentially combine multiple weak learners, usually simple decision stumps, to create a strong predictive model. Each weak learner is trained to focus on the instances that were misclassified by previous learners, thereby adaptively adjusting the model's attention toward harder-to-predict samples.

$$e_t = \sum_{i=1}^{N} \quad w_i^{(t)} (h_t(x_i) \neq y_i) \tag{17}$$

where $w_i^{(t)}$ denotes the weight of sample i at iteration t, and $h_t(x_i)$ is the prediction of the weak learner. The weight of each weak learner is computed as

$$\alpha_t = \frac{1}{2} ln\left(\frac{1-e_t}{e_t}\right) \tag{18}1$$

which reflects its contribution to the final ensemble. The sample weights are updated according to

$$w_i^{(t+1)} = \frac{w_i^{(t)} exp(-\alpha_t \gamma_i h_t(x_i))}{Z_t} \tag{19}$$

where Z_t is a normalization factor ensuring $\sum \quad w_i^{(t+1)} = 1$. Misclassified samples receive higher weights, guiding subsequent weak learners to focus on difficult instances. The last classifier is obtained by a weighted majority vote:

$$H(x) = sign\left(\sum_{t=1}^{T} \quad \alpha_t h_t(x)\right) \tag{20}$$

3.1.10. Multi-Layer Perceptron

The MLP is a feedforward neural network designed to capture nonlinear relationships between inputs and index variables, Kruse, R., et el. (2022). It is composed of an input layer, one or more hidden layers, and an output layer. Within each hidden layer, neurons carry out a linear combination of the inputs, incorporate a bias term, and apply a nonlinear activation function such as Tanh, ReLU. SiLU or to generate the output.

The learning process of an MLP is based on backpropagation, using gradient descent to optimize the weights in order to minimize the error between predicted and actual values. Due to its multi-layered structure and nonlinear activation functions, MLP can capture complex relationships in the data, making it a powerful tool for regression, classification, time series forecasting, and pattern recognition tasks. For layer l, the transformation is defined as:

$$z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)} \tag{21}$$

and the activation is applied as $a^{(l)} = f(z^{(l)})$ where $W^{(l)}$ and $b^{(l)}$ represent for weight and bias, respectively, and $f(.)$ denotes the activation function, such as ReLU, sigmoid, or tanh. Training the network involves minimizing a loss function L through backpropagation. Using the chain rule, the gradients are computed, and the derivative of the loss with respect to the weight matrix of layer l is given by Equations (22).

$$\frac{\alpha L}{\alpha W^{(l)}} = \delta^{(l)}(\alpha^{(l-1)})^T \tag{22}$$

where $\delta^{(l)}$ represents the error term propagated backward through the layers.

3.2. Local Interpretable Model-Agnostic Explanations

LIME, Ribeiro, M. T et el. (2016). provides an agnostic model-based framework for generating locally faithful explanations of individual predictions through interpretable surrogate model approximation. The methodology operates by perturbing the input instance within its local neighbourhood and sampling the complex model's behavior across these variations to construct a dataset of input-output relationships. Distance-weighted sampling ensures that perturbations closer to the original instance receive greater influence in the explanation process, maintaining local fidelity around the prediction of interest. An interpretable linear model is subsequently fitted to this weighted perturbed dataset, with feature coefficients representing the local importance and directional influence of each input variable for the specific prediction. The approach minimizes an objective function that balances explanation fidelity to the original model against the complexity of the interpretable surrogate, ensuring both accuracy and comprehensibility. Implementation generates explanations for individual demand forecasts by identifying which temporal features most significantly influence specific predictions, enabling supply chain managers to understand the underlying drivers of automated forecasting decisions at the instance level.

3.3. Dataset

In this research, we use the publicly released DataCo Supply-Chain dataset as an open benchmark to enable reproducibility, Constante, et el. (2019), which was previously utilized by DataCo Global for analytical. The deposit provides structured transactional records and logistics attributes across multiple product categories and geographies, to align the benchmark with e-commerce use, we aggregate transactions to the daily level and engineer features commonly available to online retailers: lagged sales, multi-day moving averages, calendar factors (day-of-week, month, holiday proximity), and simple promotion/price intensity proxies derived from discount fields. The dataset contains detailed records of multi-product supply chain management over the period from January 19, 2016, to February 3, 2018, with 180,519 observations and 53 variables.

The target variable selected for this study is Sales. Descriptive statistics indicate: count = 180,519; mean = 203.772096; minimum = 9.99; maximum = 1999.989990; standard deviation = 132.273077. The distribution of Sales demonstrates substantial variability across orders, suggesting the presence of heterogeneity and outliers within the data. Consequently, it is necessary to apply forecasting models capable of effectively handling data dispersion and imbalance.

3.4. Data Preprocessing

The raw dataset was first examined to identify key variables, their data types, and the presence of missing values. Two variables, Product Description and Order Zipcode, exhibited extremely high rates of missing values and were therefore removed to ensure data quality. The order date variable was converted to a datetime format, and the dataset was sorted chronologically. The data were then aggregated on a daily basis by summing Sales across all Product Category Id, which helped standardize the data scale and reduce noise.

During the feature engineering stage, lag features Sales_Lag1 and Sales_Lag2 were generated from Sales to allow models to learn from previous days' sales. Moving average features Sales_MovingAvg_3 and Sales_MovingAvg_7 were constructed to capture short-term and weekly trends. Additionally, temporal components Day, Month, and Year were extracted from order date (DateOrders) to help models recognize seasonal patterns. Rows with missing values resulting from the creation of lag or moving average features got deleted to maintain data integrity.

After the process, we split the dataset into training sets and sets using an 80/20 chronological split. The input features (X) included Sales_Lag1, Sales_Lag2, Sales_MovingAvg_3, Sales_MovingAvg_7, Day, Month, and Year, while the target variable (y) was Sales. This process produced a clean, well-structured, and temporally enriched dataset, suitable for the application of both machine learning and time series models, enabling the capture of trends and seasonal patterns in sales.

4. RESULTS AND EXPLAINATIONS

4.1 Evaluation Results

In this research, evaluations results were outlined as in Table (1). LSTM achieves superior performance (RMSE: 42.20, R^2: 98.79%, MAPE: 0.41%), demonstrating the critical importance of temporal modeling for supply chain forecasting. XGBoost provides the best traditional ML alternative with RMSE of 154.58, and R^2: 83.71%. Tree-based ensembles exhibit metric inconsistencies with anomalously high R^2 values (>99%) despite poor RMSE performance, suggesting potential overfitting issues. Traditional approaches with SVM of 1419.68 RMSE, MLP of 1639.46 RMSE fail to capture temporal dependencies effectively. These findings validate that temporal modeling capabilities are essential for e-commerce demand forecasting, with LSTM offering optimal accuracy-interpretability balance for practical deployment, furthermore Figure 2 illustrated the forecasting results of the LSTM model on the dataset.

Table 1. Evaluations Result Across Ten Models

ID	Models	RMSE ↓	R2 ↑	MAPE ↓
1	XGBoost	154.5788	83.71%	1.7652
2	Random Forest	266.2974	99.36%	10.4283
3	Gradient boosting	442.9231	97.60%	26.2422
4	Decision tree	63.0383	77.08%	35.66%
5	Catboost	224.2307	99.55%	6.8143
6	SVM	1419.6808	82.42%	161.4589
7	Lightgbm	240.2319	99.48%	8.8718
8	ADABoost	903.9515	90.80%	100.8299
9	MLP	1639.4611	77.11%	114.7144
10	LSTM	42.2024	98.79%	0.4154

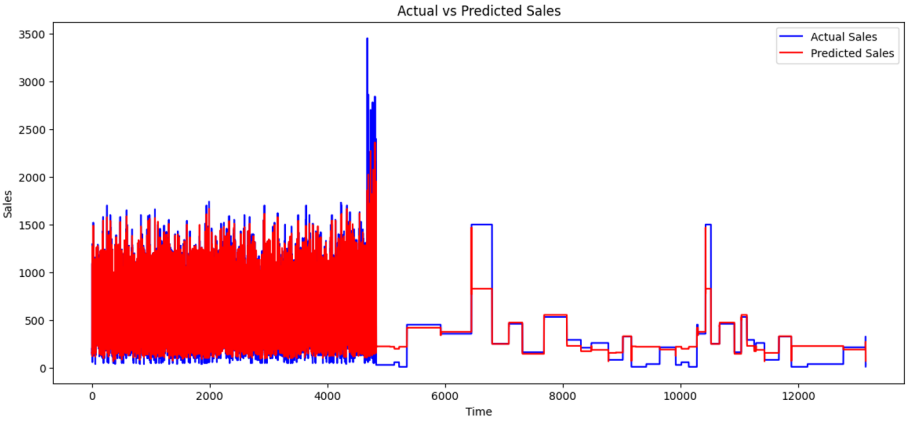

Fig. 2. LSTM model Forecasting Results

4.2. Daily-Level Alignments

We generate explanations at the daily forecast level to match the modeling granularity, as in Figure 3 provides a local interpretation of the model's prediction for a single transaction. The strongest negative contributions originate from discount-related

attributes, namely Order Item Discount (> 0.37) and Order Item Discount Rate (> 0.83), which substantially reduce the predicted outcome. This pattern suggests that aggressive pricing strategies are consistently associated with lower performance scores. Profitability indicators, such as Order Profit per Order (≤ −0.27) and Order Item Profit Ratio (≤ −0.98), further depress the prediction, underscoring the decisive role of revenue margins in shaping the model's assessment. Logistics variables exhibit a dual influence. Extended delivery times, captured by Days for shipping (real) ≤ −0.92, exert a marked negative effect, whereas early planned shipments (Days for shipment (scheduled) ≤ −0.68) act as salient positive drivers, reflecting the operational value of timely scheduling.

A secondary group of predictors produces moderate adjustments. Customer-related measures (*e.g.*, Sales per customer) and categorical identifiers such as *Product Card Id* introduce small yet discernible shifts, pointing to nuanced interactions between purchasing behaviour and item categorisation. By contrast, features like *Order Id* and *Cardprod Id* retain measurable coefficients without clear business semantics; their explanatory weight likely stems from indexing artefacts, implying that further feature-engineering or variable screening is warranted. Overall, the LIME explanation elucidates that the model's local decision is primarily informed by pricing policy, profitability structure, and delivery planning, with minor refinements provided by customer segmentation and product labelling. These findings reinforce the managerial importance of balanced discount policies and efficient shipping schedules in sustaining order performance predictions.

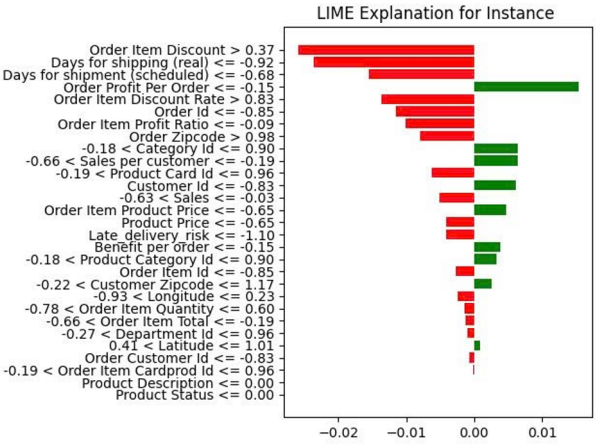

Fig. 3. Lime Analysis from the dataset

5. CONCLUSION

In this research, we compared the performance between many architectures in supply chain demand regression with xAI tools to enhance transparency. Experimental results showing the LSTM model outperforms robust boosting algorithms, confirming the pivotal role of capturing sequential dependencies in time series data. The integrated framework between the high-performance LSTM model and explanation methods such as LIME has shown great potential in improving both forecast accuracy and confidence in management decision making. Although the study is limited by evaluating only a single dataset, future research directions should focus on validating this method on more diverse datasets and comparing it with advanced architectures such as Transformer.

6. ACKNOWLEDGMENTS

This research is supported by International School, Vietnam National University, Hanoi (VNU-IS). Thanks CoMI group for computational unit usage.

REFERENCES

Wu, S., Zhang, Z., & Ru, Y. (2024). Research on product demand forecasting based on random forest and arima time series: Precision forecasting method for data-scarce environments. Transactions on Computer Science and Intelligent Systems Research, 5. https://doi.org/10.62051/g9r9ca46

Torres, D., & Cepeda, L. (2024). Machine learning for predicting online shoppers' purchase intentions. https://doi.org/10.13140/RG.2.2.15297.36968

OECD. (2019). Unpacking e-commerce: Business models, trends and policies. OECD Publishing. https://doi.org/10.1787/23561431-en

Li, J., Cui, T., Yang, K., Yuan, R., He, L., & Li, M. (2021). Demand Forecasting of E-Commerce Enterprises Based on Horizontal Federated Learning from the Perspective of Sustainable Development. Sustainability, 13(23), 13050. https://doi.org/10.3390/su132313050

Jahin, M. A., Shahriar, A., & Amin, M. A. (2025). MCDFN: supply chain demand forecasting via an explainable multi-channel data fusion network model. Evolutionary Intelligence, 18(3). https://doi.org/10.1007/s12065-025-01053-7

Kim, J., Kim, H., Kim, H., Lee, D., & Yoon, S. (2025). A comprehensive survey of deep learning for time series forecasting: architectural diversity and open challenges. Artificial Intelligence Review, 58(7). https://doi.org/10.1007/s10462-025-11223-9

Moktadir, M. A., Ali, S. M., Paul, S. K., & Shukla, N. (2019). Barriers to big data analytics in manufacturing supply chains: A case study from Bangladesh. Computers & Industrial Engineering, 128, 1063–1075. https://doi.org/10.1016/j.cie.2018.04.013

Zhang, Q. (2025). Sparse and intermittent demand forecasting in supply chains: A model-router framework. arXiv preprint. https://doi.org/10.48550/arXiv.2506.14810

Douaioui, K., Oucheikh, R., Benmoussa, O., & Mabrouki, C. (2024). Machine Learning and Deep Learning Models for Demand Forecasting in Supply Chain Management: A Critical Review. Applied System Innovation (ASI), 7(5), 93. https://doi.org/10.3390/asi7050093

Balan, G. S., Kumar, V. S., & Raj, S. A. (2025). Machine learning and artificial intelligence methods and applications for post-crisis supply chain resiliency and recovery. Supply Chain Analytics, 10, 100121. https://doi.org/10.1016/j.sca.2025.100121

Chen, T., & Guestrin, C. (2016, August). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785

Mitchell, R., Adinets, A., Rao, T., & Frank, E. (2018). XGBoost: Scalable GPU accelerated learning. arXiv preprint. https://doi.org/10.48550/arXiv.1806.11248

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Salman, H. A., Kalakech, A., & Steiti, A. (2024). Random forest algorithm overview. Babylonian Journal of Machine Learning, 2024, 69-79. https://doi.org/10.58496/BJML/2024/007

Friedman, J. (2001). Greedy boosting approximation: a gradient boosting machine. Ann. Stat. 29, 1189–1232. doi: 10.1214/aos/1013203451

Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1, 81–106. https://doi.org/10.1007/BF00116251

Dorogush, A. V., Ershov, V., & Gulin, A. (2018, October). CatBoost: gradient boosting with categorical features support. arXiv preprint. https://doi.org/10.48550/arXiv.1810.11363

Schölkopf, B., Smola, A. J., Williamson, R. C., & Bartlett, P. L. (2000). New support vector algorithms. Neural Computation, 12(5), 1207-1245. https://doi.org/10.1162/089976600300015565

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30. https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html?utm_source=chatgpt.com

Freund, Y., & Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. In icml (Vol. 96, pp. 148-156). http://dl.acm.org/citation.cfm?id=3091696.3091715

Kruse, R., Mostaghim, S., Borgelt, C., Braune, C., & Steinbrecher, M. (2022). Multi-layer perceptrons. In Computational intelligence (Texts in Computer Science). Springer, Cham. https://doi.org/10.1007/978-3-030-42227-1_5

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135-1144. https://doi.org/10.1145/2939672.2939778

Constante, Fabian; Silva, Fernando; Pereira, António (2019), "DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS", Mendeley Data, V5, https://doi.org/10.17632/8gx2fvg2k6.