Time Series Forecasting with GCN-LSTM Based Unified Model for Product Demand Prediction

Melike Yildiz Aktas*, Taoran Ji‡, and Chang-Tien Lu*
*Department of Computer Science, Virginia ‡ Texas A&M University-Corpus Christi
{melike@vt.edu, taoran.ji@tamucc.edu, clu@vt.edu}

2024 IEEE International Conference on Big Data (BigData) | 979-8-3503-6248-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/BigData62323.2024.10825969

***Abstract*—This paper introduces LSTMGraph, a unified time-series model designed for demand prediction across multiple products. This method integrates Long Short-Term Memory (LSTM) networks to capture temporal dynamics, such as price fluctuations, and Graph Convolutional Networks (GCN) to model global dependencies between products. We represent demand data as a network where each product is a node, constructing three distinct graphs with different types of edges: (i) a weekly sales similarity graph, (ii) a customer-based relationship graph, and (iii) an invoice-based similarity graph. These graphs are merged to enhance predictive accuracy by incorporating diverse temporal and relational patterns. Extensive experiments show that LSTMGraph significantly outperforms existing baseline models. Additionally, an ablation study is conducted to quantify the impact of each graph type on overall performance.**

***Index Terms*—Demand prediction, Long short-term memory, Graph convolutional network, Time series forecasting**

I. Introduction

Reducing holding costs is a critical priority for manufacturers, as excessive inventory directly impacts profitability [1]. Accurate demand forecasting—whether on a daily, weekly, monthly, or annual basis—has become essential for optimizing production and supply chain management [2]. Achieving reliable forecasts requires a thorough analysis of product characteristics, temporal patterns, and customer demand behavior, all of which play a crucial role in aligning inventory with actual market needs [3]. Demand prediction remains a critical and evolving research area due to its significant role across various industries, including pharmaceutical supply chains [4] and grocery retailing [2]. Accurate forecasting in these domains helps optimize inventory management, reduce costs, and meet fluctuating customer demands, highlighting the importance of advanced predictive models in complex supply chain systems.

There are numerous techniques for demand prediction in the literature. Traditional forecasting methods, often based on time-series models, rely heavily on limited historical data [2], which can result in significant information loss and reduced predictive accuracy. These approaches may struggle to capture complex temporal patterns or relationships between different products and external factors, limiting their effectiveness in dynamic, real-world applications. Today, machine learning and data mining techniques are increasingly being applied to time series forecasting, enabling the capture of more intricate patterns and relationships [5]. These advanced methods allow for the integration of larger datasets, overcoming the limitations of traditional approaches and improving the accuracy of demand predictions by leveraging complex temporal and contextual information.

In this paper, we propose a novel model to enhance the demand forecasting process by leveraging a dataset that captures the customer-product-sales relationship. To the best of our knowledge, few studies [5], [6] have explored this tripartite relationship in depth. Our approach combines LSTM networks, which effectively model temporal dependencies, with GCNs, allowing us to capture global relationships between products. This integration enables more accurate demand predictions by accounting for both time-based trends and complex product interactions.

We summarize our contribution as follows:

- **Time-series forecasting modeling (LSTMGraph):** We present a predictive modeling technique that employs Long Short-Term Memory (LSTM) networks and Graph Convolutional Networks (GCN) in a unified framework. This approach leverages the sequential and temporal relationships of products to accurately predict their demands.
- **Analyzing a comprehensive customer-product-sales dataset:** We introduce a method for creating various types of graphs using the same dataset, enhancing our analysis of customer interactions and product sales dynamics.
- **Conducting extensive experiments:** We perform experiments on the dataset to demonstrate that our model effectively captures both local and global associations, leading to improved forecasting accuracy.
- **Evaluation of model performance:** We assess the performance of our LSTMGraph model using standard metrics, comparing it with existing forecasting methods to validate its effectiveness and robustness in real-world applications.

II. Related Work

In this section, three groups of related work are discussed: 1) Demand Prediction 2) LSTM for Time Series Problem and 3) Graph Neural Network for Time Series Problem.

A. Demand Prediction

Kumar et al. [7] employed a back-propagation neural network model trained with fuzzy inputs, utilizing historical demand and sales data alongside advertising effectiveness, expenditures, promotions, and marketing event data to enhance prediction accuracy. In a complementary approach, Kilimci et al. [2] proposed an integration strategy that combines multiple

979-8-3503-6248-0/24/$31.00 ©2024 IEEE

forecasting models to improve demand prediction outcomes. Abbasimehr et al. [8] introduced a demand forecasting method leveraging multi-layer Long Short-Term Memory (LSTM) networks, demonstrating the effectiveness of deep learning in capturing complex demand patterns. Huber and Stuckenschmidt [9] focused on a bakery chain, specifically addressing the challenge of forecasting daily demand for various product categories, particularly around special calendar events. Meanwhile, Nguyen et al. [10] developed a data-mining prediction framework for remanufactured products, exploring the nonlinear relationships of online market factors as predictors of customer demand. Li et al. [11] introduced SGNN that utilizes the Graph Neural Network (GNN) for demand prediction by leveraging the spatial relationships inherent in online sales data.

B. LSTM for Time Series Problem

Abbasimehr and Paki [12] integrated Long Short-Term Memory (LSTM) networks with attention mechanisms to enhance time-series prediction across various datasets. In a focused application, Dubey et al. [13] utilized LSTM for forecasting energy consumption, comparing its performance against traditional models such as ARIMA and seasonal ARIMA (SARIMA). Livieris et al. [14] combined Convolutional Neural Networks (CNN) and LSTM to achieve accurate predictions of gold price movements. Chimmula and Zhang [15] developed a real-time forecasting model using LSTM networks to predict COVID-19 transmission dynamics. Ning et al. [16] conducted a comparative analysis of ARIMA, LSTM, and Prophet models, revealing that both ARIMA and LSTM outperformed Prophet in oil price prediction tasks. Zhang et al. [17] proposed a deep learning hybrid prediction model for stock market forecasting, incorporating Complementary Ensemble Empirical Mode Decomposition (CEEMD), Principal Component Analysis (PCA), and LSTM to leverage their complementary strengths. Yadav et al. [18] also employed LSTM for stock market predictions, highlighting its effectiveness in capturing market trends. Furthermore, Hu et al. [19] introduced a variant of LSTM that enhances the memory module by merging the forget and input gates into a single update gate, utilizing a Sigmoid layer to regulate information flow.

C. Graph Neural Networks for Time Series Problem

Lazcano et al. [20] integrated Graph Convolutional Networks (GCNs) with Bidirectional Long Short-Term Memory (BiLSTM) networks to effectively predict oil prices. Wu et al. [21] introduced a general graph neural network framework specifically tailored for multivariate time series data, which adeptly extracts unidirectional relationships among variables. Cao et al. [22] developed StemGNN, a model that simultaneously captures inter-series correlations and temporal dependencies in the spectral domain. Cheng et al. [23] proposed a Multi-modality Graph Neural Network (MAGNN) designed to learn from multimodal inputs for financial time series predictions. Wang et al. [24] combined graph neural networks with multivariate time series (MTS) analysis to uncover the internal temporal patterns of single-dimensional time series while considering the complex spatial relationships among variables. Zanfei et al. [25] created a Graph Convolutional Recurrent Neural Network (GCRNN) to forecast water demand time series for water supply systems. Khodayar and Wang [26] proposed a spatio-temporal graph deep neural network aimed at short-term wind speed forecasting. Additionally, Cui et al. [27] introduced the Traffic Graph Convolutional Long Short-Term Memory Neural Network (TGC-LSTM), which captures interactions among roadways in the traffic network to forecast network-wide traffic states. Diao et al. [28] also contributed a dynamic spatio-temporal Graph Convolutional Neural Network (GCNN) for accurate traffic forecasting.

III. METHODOLOGY

In this study, we aim to accurately define the dependencies and perform time-series forecasting for multiple product quantities by analyzing a comprehensive set of features. Our methodology is structured around three core components: robust feature extraction, creating different types of graphs, and strategic LSTM and GCN integration. Each component uniquely contributes to meeting the predictive demands of our model. A visual representation of our framework is illustrated in Figure 1.

A. Preprocessing and Feature Extraction

We adopted a cleaning process similar to that outlined by Yilmaz et al. [6]. In our dataset, we removed all transactions lacking a description or customer ID, as well as any cancelled transactions. We also excluded transactions with product prices below zero, as these are not feasible in real-world scenarios. Additionally, stock codes containing letters were deemed irrelevant for this study and were consequently discarded. Analysis of the price feature revealed that the 75th percentile is 3.75, while the maximum value is 649.5, indicating the presence of outliers. To address this, we applied a logarithmic transformation to the price feature and eliminated transactions with outlier UnitPrice values, retaining only those that fall within the top 99% of the data distribution. We implemented the same procedure for the Quantity feature.

Furthermore, we created daily transaction records for each product. Notably, some products exhibited zero demand for over 40% of the two-year period, leading us to exclude them to enhance prediction accuracy. We also observed instances of daily demand exceeding 200 units for certain products, which were considered excessive. To ensure a balanced dataset, we removed these outlier transactions as well. From the transaction data, we generated new columns to capture the minimum and maximum prices of products for each daily transaction. Additionally, we extracted features from the invoice date, including columns for the day, day of the week, month, and year.

B. Graph Generation

We conceptualize products as nodes within a network, with their connections representing various interrelationships. To

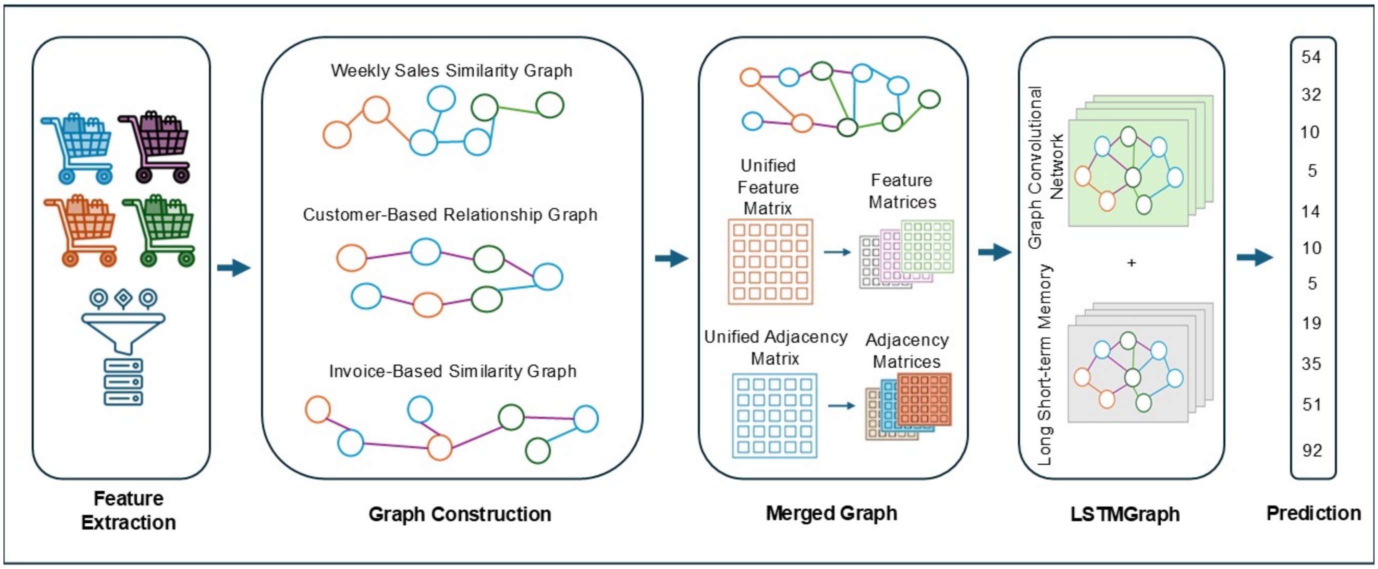

Fig. 1. Proposed Framework

get different understanding of data, we created three types of graph. In the graph construction phase, we define three distinct graph structures, denoted as $G_m = (V, E_m)$, where m specifies the graph (a weekly sales similarity graph (G1), a customer-based similarity graph (G2), and invoice-based relationship graph (G3) and each graph contains a set of n nodes $V = \{v_1, v_2, \ldots, v_n\}$. The adjacency matrix for each graph is represented as $A_m \in \mathbb{R}^{n \times n}$, and the node feature matrix is indicated by $X \in \mathbb{R}^{n \times F}$, where F represents the number of features for each node.

For weekly sales similarity graph, we counted how many times two products are sold more than 50 in a week and divided this to 54 (number of weeks in a year) to get edge e_{ij}^m between nodes v_i and v_j and edge weights. In customer-based graph, the nodes that are sold more than 5 times together by a customer are connected to each other. For invoice-based relationship graph, edges are determined if two products are together in an invoice. The ultimate objective is to use these graph structures to predict demand quantities for different products, represented by $Y = \{Y_1, Y_2, \ldots, Y_t\}$, where t specifies day information. We decide thresholds to get a connected graph.

C. Integration of LSTM and GCN

A sliding window approach is used to extract time-series data of product purchases. Given the time-series data for n products and t days, we use a sliding window of size w (e.g., 15 days) with a stride s (e.g., 1 day) to create overlapping windows of product features. This approach captures both short-term and long-term demand fluctuations.

For each window, the product features are extracted from the original dataset and additional features such as minimum price, maximum price, and clustering coefficient that comes from weekly sales similarity graph are included, resulting in a rich feature set:

$$X_{\text{new}} = X_{\text{quantity}} + X_{\text{minPrice}} + X_{\text{maxPrice}} + X_{\text{coefficient}} \quad (1)$$

If a sliding window at the end of the dataset has fewer than w time steps, zero-padding is applied to maintain consistency in window size across the dataset. The input features X are converted to tensors and paired with the edge index to create a graph representation for each window.

We combined adjacency matrices by concatenating them to get a unified matrix.The new adjacency matrix is updated as:

$$A_{\text{new}} = (A_1 + A_2 + A_3) \quad (2)$$

Given the merged graph G with node feature matrix $X_{\text{new}} \in \mathbb{R}^{n \times (F)}$ and the combined adjacency matrix A_{new}, the GCN operates as follows:

$$H = A_{\text{new}} X_{\text{new}} \Theta \quad (3)$$

where $\widetilde{A}_{\text{new}}$ is the normalized version of A_{new}, calculated by:

$$\widetilde{A}_{\text{new}} = D^{-\frac{1}{2}} A_{\text{new}} D^{-\frac{1}{2}} + I \quad (4)$$

with D representing the diagonal node degree matrix and I the identity matrix. Each GCN layer i updates as:

$$H^i = \widetilde{A}_{\text{new}} H^{i-1} \Theta \quad (5)$$

and the output from the final layer is computed by:

$$H^{\text{graph}} = \sigma(H^l \Theta) \quad (6)$$

Here, σ represents the ReLU activation function, Θ learnable parameter, and l the number of hidden layers.

Once the GCN layer has processed the graph structure, we keeped the output feature matrix H^{graph}.

To capture temporal dependencies, the LSTM processes the sequential information for each node as follows:

For each time step t, the LSTM takes the hidden state h_{t-1}, cell state c_{t-1}, and the input H_t from the beginning:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, H_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, H_t] + b_i) \\ \tilde{c}_t &= \tanh(W_c \cdot [h_{t-1}, H_t] + b_c) \\ c_t &= f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \\ o_t &= \sigma(W_o \cdot [h_{t-1}, H_t] + b_o) \\ h_t &= o_t \cdot \tanh(c_t) \end{aligned} \tag{7}$$

Here:

- f_t is the forget gate,
- i_t is the input gate,
- $\tilde{c}_t$ is the candidate cell state,
- c_t is the updated cell state,
- o_t is the output gate, and
- h_t is the updated hidden state, which is passed to the next time step.

After passing through all the time steps, the final hidden state h_T from the LSTM layer is the output. By taking average the outputs from both GCN and LSTM layers, we derive the final predictions H_{final} for demand prediction task, effectively capturing both local and global structural information.

$$H_{\text{final}} = \frac{h_T + H_{\text{graph}}}{2} \tag{8}$$

To train the model, L1 loss for Mean Absolute Error (MAE) is defined as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^{n} (y_i - H_{\text{final},i})^2 \tag{9}$$

where y_i represents the true demand value for the i-th sample, and $H_{\text{final},i}$ represents the predicted demand value for the i-th sample.

IV. Experiments

To assess the performance of our proposed model, LSTMGraph, we conducted extensive experiments to validate its effectiveness.

A. Dataset

In this study, we utilize a publicly available dataset [29] from an online retail company, covering two-year period. The dataset contains detailed records of customer transactions, including the following attributes: invoice number, stock code, product description, quantity purchased, invoice date, unit price, customer ID, and the country of origin. This comprehensive dataset provides a rich source of information for analyzing purchase behavior and demand patterns across different products and geographic regions as can be seen in Table I. It comprises a total of 1,067,371 transactions, spread across 53,628 unique invoices. These transactions involve 5,305 unique products, which are described by 5,698 unique product descriptions. The dataset also includes records for 5,942 unique customers from 43 different countries. For each transaction, details such as quantity and price are provided, which vary across different entries. Additionally, the invoice date captures temporal aspects with information on the hour, day, month, and year, allowing for in-depth temporal analysis of purchasing behavior. This rich and diverse dataset provides an extensive basis for studying transaction patterns, customer behavior, and other significant insights within a global context.

TABLE I
DATASET DESCRIPTION

Features	Details
Transactions	1067371
Unique Invoice	53628
Unique Products	5305
Unique Description	5698
Unique Customer ID	5942
Unique Country	43
Quantity	Different for Each Transaction
Price	Different for Each Transaction
InvoiceDate	Includes Hour, Day, Month, Year Information

B. Baselines

We compared our proposed method with 7 baselines that can be categorized baselines into two groups: traditional time-series and machine learning methods.

Traditional methods:

- Moving Average [30]: A simple method that smooths past demand by averaging observations over a fixed number of time steps.
- Exponential Moving Average [30]: A weighted moving average that gives more importance to recent observations to predict future demand.
- ARIMA [31]: A statistical model that uses past values and errors in a linear manner to predict future points in the time series.

Machine learning methods:

- Long Short-term Memory [32]: A type of recurrent neural network designed to remember long-term dependencies.
- Gated Recurrent Units [33]: A type of recurrent neural network (RNN) that simplifies the architecture of LSTM by combining the forget and input gates.
- Graph Convolutional Network [34]: A type of graph neural network that aggregates feature information from a node's local neighborhood to generate new node representations, effectively capturing the structural information of the graph.
- Graph Attention Network [35]: A type of graph neural network that uses an attention mechanism to assign different importance levels to neighboring nodes, allowing the network to focus on more relevant nodes during aggregation.

C. Evaluation Metrics and Experiment Settings

LSTMGraph utilizes historical data from the previous 15 days to forecast product demand, employing a stride of 1 for the analysis. The dataset is split into training and testing subsets, with 80% allocated for training and 20% reserved

TABLE II
COMPARISON OF DIFFERENT METHODS AND THEIR CORRESPONDING MEAN ABSOLUTE ERRORS (MAE).

Models	Mean Absolute Error (MAE)
Moving Average	12.7859
Exponential Moving Average	12.3466
ARIMA	13.9902
SARIMA	14.4505
GCN	12.3381
Graph Attention Network	12.2699
LSTM	12.2828
Gated Recurrent Units	13.1009
LSTMGraph	11.4776

for testing. To evaluate the performance of our model, we employed the Mean Absolute Error (MAE) metric, which provides a clear measure of the average prediction error.

V. RESULTS

A. Demand Prediction Results

The results of the experiments comparing various forecasting methods are summarized in Table II, where each model's performance is evaluated based on the Mean Absolute Error (MAE). Among the tested models, traditional statistical methods such as Moving Average and Exponential Moving Average achieved MAEs of 12.7859 and 12.3466, respectively, showing reasonable predictive capabilities. The ARIMA and SARIMA models yielded higher MAEs of 13.9902 and 14.4505, indicating their limitations in capturing complex dependencies in the dataset.

Deep learning models demonstrated improved performance compared to statistical methods. The Graph Convolutional Network (GCN) and Graph Attention Network (GAT) achieved MAEs of 12.3381 and 12.2699, respectively, reflecting their ability to leverage graph structures effectively. LSTM and Gated Recurrent Units (GRU) also performed well, with MAEs of 12.2828 and 13.1009, suggesting their strengths in capturing temporal patterns. However, the LSTMGraph model, which integrates both temporal and graph-based features, achieved the lowest MAE of 11.4776, demonstrating its superior efficacy in combining sequential and structural information for improved predictive accuracy.

These results highlight the efficacy of the LSTMGraph model in providing more accurate predictions compared to both traditional statistical models and other deep learning approaches. The significantly lower MAE indicates that LSTMGraph successfully captures both temporal dependencies and graph-based relationships, outperforming existing methods in the context of this task.

B. Ablation for Graph Types

The ablation study results, presented in Table III, provide insights into the impact of different graph structures on the performance of the LSTMGraph model, evaluated using Mean Absolute Error (MAE). We considered three types of graph structures: Weekly Sales Similarity Graph, Customer-based Relationship Graph, and Invoice-based Similarity Graph. The study systematically analyzes the performance with different combinations of these graph structures, highlighting their contributions to the overall predictive accuracy.

TABLE III
COMPARISON OF DIFFERENT GRAPH STRUCTURES AND THEIR CORRESPONDING MEAN ABSOLUTE ERRORS (MAE).

G1	G2	G3	Mean Absolute Error (MAE)
+			11.6834
	+		12.3293
		+	11.8377
+	+		11.9633
	+	+	11.9417
+		+	11.9802

The Weekly Sales Similarity Graph alone resulted in the lowest MAE of 11.6834, indicating that leveraging weekly sales patterns is particularly effective for the task. When using the Invoice-based Similarity Graph alone, the model achieved an MAE of 11.8377, showing a strong but slightly less effective contribution compared to the Weekly Sales Similarity graph. The Customer-based Relationship Graph alone produced a higher MAE of 12.3293, suggesting that customer relationships, while useful, may not be as predictive in isolation as the other graph types.

Combining different graphs provided mixed results. The combination of Weekly Sales Similarity and Customer-based Relationship Graphs led to an MAE of 11.9633, while combining Customer-based Relationship with Invoice-based Similarity Graphs resulted in an MAE of 11.9417. Additionally, using both Weekly Sales Similarity and Invoice-based Similarity Graphs produced an MAE of 11.9802. Notably, none of these combinations outperformed the use of the Weekly Sales Similarity Graph by itself, suggesting that the additional information from other graphs may introduce redundancy or noise.

Overall, the ablation study reveals that the Weekly Sales Similarity Graph is the most influential graph structure for minimizing prediction error. The results highlight the importance of carefully selecting graph structures, as incorporating multiple graphs does not always lead to better performance. The findings suggest that focusing on key structural relationships, such as those represented by weekly sales similarity, can be more effective for optimizing the model's predictive capability.

VI. CONCLUSION

In this paper, we proposed the LSTMGraph model, a novel approach that integrates both temporal and structural graph features for improved forecasting accuracy in dynamic graph-based datasets. Our method effectively combines the strengths of Long Short-Term Memory (LSTM) networks for capturing temporal dependencies with graph-based techniques to exploit relational structures inherent in the data. Through extensive experiments, we demonstrated the superiority of LSTMGraph over traditional statistical methods and other deep learning models, achieving the lowest Mean Absolute Error (MAE) of 11.4776, indicating a significant improvement in predictive performance.

The LSTMGraph model's ability to handle both temporal and structural components, combined with its scalability and adaptability to different graph structures, makes it well-suited for a wide range of real-world applications involving dynamic and evolving graphs. Future work could explore further enhancements by incorporating more sophisticated graph aggregation techniques or by extending the model to multi-task learning scenarios where additional graph information could be leveraged. Overall, the proposed LSTMGraph framework offers a powerful and flexible solution for time-series forecasting tasks in graph-structured environments, achieving both high accuracy and robustness.

Acknowledgment

Melike Yildiz Aktas is financially supported by the Turkish Ministry of National Education for her PhD research.

References

[1] O. Torkul, R. Yılmaz, İ. H. Selvi, and M. R. Cesur, "A real-time inventory model to manage variance of demand for decreasing inventory holding cost," *Computers & Industrial Engineering*, vol. 102, pp. 435–439, 2016.

[2] Z. H. Kilimci, A. O. Akyuz, M. Uysal, S. Akyokus, M. O. Uysal, B. Atak Bulbul, M. A. Ekmis *et al.*, "An improved demand forecasting model using deep learning approach and proposed decision integration strategy for supply chain," *Complexity*, vol. 2019, 2019.

[3] G. Karpushkin, "Predicting consumer behavior based on big data of user-generated online content in retail marketing," *Global Journal of Flexible Systems Management*, vol. 25, no. 1, pp. 163–178, 2024.

[4] G. Merkuryeva, A. Valberga, and A. Smirnov, "Demand forecasting in pharmaceutical supply chains: A case study," *Procedia Computer Science*, vol. 149, pp. 3–10, 2019.

[5] V. Taparia, P. Mishra, N. Gupta, and D. Kumar, "Improved demand forecasting of a retail store using a hybrid machine learning model," *Journal of Graphic Era University*, pp. 15–36, 2024.

[6] Y. Yilmaz and Ş. G. ÖĞüdücü, "Enriching demand prediction with product relationship information using graph neural networks," in *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2021, pp. 561–568.

[7] A. Kumar, R. Shankar, and N. R. Aljohani, "A big data driven framework for demand-driven forecasting with effects of marketing-mix variables," *Industrial marketing management*, vol. 90, pp. 493–507, 2020.

[8] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using lstm network for demand forecasting," *Computers & industrial engineering*, vol. 143, p. 106435, 2020.

[9] J. Huber and H. Stuckenschmidt, "Daily retail demand forecasting using machine learning with emphasis on calendric special days," *International Journal of Forecasting*, vol. 36, no. 4, pp. 1420–1438, 2020.

[10] T. Van Nguyen, L. Zhou, A. Y. L. Chong, B. Li, and X. Pu, "Predicting customer demand for remanufactured products: A data-mining approach," *European Journal of Operational Research*, vol. 281, no. 3, pp. 543–558, 2020.

[11] J. Li, L. Fan, X. Wang, T. Sun, and M. Zhou, "Product demand prediction with spatial graph neural networks," *Applied Sciences*, vol. 14, no. 16, p. 6989, 2024.

[12] H. Abbasimehr and R. Paki, "Improving time series forecasting using lstm and attention models," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 1, pp. 673–691, 2022.

[13] A. K. Dubey, A. Kumar, V. García-Díaz, A. K. Sharma, and K. Kanhaiya, "Study and analysis of sarima and lstm in forecasting time series data," *Sustainable Energy Technologies and Assessments*, vol. 47, p. 101474, 2021.

[14] I. E. Livieris, E. Pintelas, and P. Pintelas, "A cnn–lstm model for gold price time-series forecasting," *Neural computing and applications*, vol. 32, pp. 17 351–17 360, 2020.

[15] V. K. R. Chimmula and L. Zhang, "Time series forecasting of covid-19 transmission in canada using lstm networks," *Chaos, solitons & fractals*, vol. 135, p. 109864, 2020.

[16] Y. Ning, H. Kazemi, and P. Tahmasebi, "A comparative machine learning study for time series oil production forecasting: Arima, lstm, and prophet," *Computers & Geosciences*, vol. 164, p. 105126, 2022.

[17] Y. Zhang, B. Yan, and M. Aasma, "A novel deep learning framework: Prediction and analysis of financial time series using ceemd and lstm," *Expert systems with applications*, vol. 159, p. 113609, 2020.

[18] A. Yadav, C. Jha, and A. Sharan, "Optimizing lstm for time series prediction in indian stock market," *Procedia Computer Science*, vol. 167, pp. 2091–2100, 2020.

[19] J. Hu, X. Wang, Y. Zhang, D. Zhang, M. Zhang, and J. Xue, "Time series prediction method based on variant lstm recurrent neural network," *Neural Processing Letters*, vol. 52, pp. 1485–1500, 2020.

[20] A. Lazcano, P. J. Herrera, and M. Monge, "A combined model based on recurrent neural networks and graph convolutional networks for financial time series forecasting," *Mathematics*, vol. 11, no. 1, p. 224, 2023.

[21] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 753–763.

[22] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong *et al.*, "Spectral temporal graph neural network for multivariate time-series forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17 766–17 778, 2020.

[23] D. Cheng, F. Yang, S. Xiang, and J. Liu, "Financial time series forecasting with multi-modality graph neural network," *Pattern Recognition*, vol. 121, p. 108218, 2022.

[24] Y. Wang, Z. Duan, Y. Huang, H. Xu, J. Feng, and A. Ren, "Mthetgnn: A heterogeneous graph embedding framework for multivariate time series forecasting," *Pattern Recognition Letters*, vol. 153, pp. 151–158, 2022.

[25] A. Zanfei, B. M. Brentan, A. Menapace, M. Righetti, and M. Herrera, "Graph convolutional recurrent neural networks for water demand forecasting," *Water Resources Research*, vol. 58, no. 7, p. e2022WR032299, 2022.

[26] M. Khodayar and J. Wang, "Spatio-temporal graph deep neural network for short-term wind speed forecasting," *IEEE Transactions on Sustainable Energy*, vol. 10, no. 2, pp. 670–681, 2018.

[27] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2019.

[28] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 890–897.

[29] V. Ummadisetty. (2017) Online retail data set. [Online]. Available: https://www.kaggle.com/datasets/vijayuv/onlineretail

[30] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *International journal of forecasting*, vol. 20, no. 1, pp. 5–10, 2004.

[31] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[32] S. Hochreiter, "Long short-term memory," *Neural Computation MIT-Press*, 1997.

[33] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[34] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[35] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.