Article

Journal of Infrastructure, Policy and Development 2024, 8(14), 9683.
https://doi.org/10.24294/jipd9683

Performance comparison of machine learning and deep learning models for supply chain tier order quantity prediction: Emphasis on tree-based and CNN-BILSTM approaches

Kyoungjong Park

Department of Business Administration, Gwangju University, Gwangju 61743, Korea; kjpark@gwangju.ac.kr

CITATION

Park K. (2024). Performance comparison of machine learning and deep learning models for supply chain tier order quantity prediction: Emphasis on tree-based and CNN-BILSTM approaches. Journal of Infrastructure, Policy and Development. 8(14): 9683. https://doi.org/10.24294/jipd9683

ARTICLE INFO

Received: 17 October 2024
Accepted: 13 November 2024
Available online: 22 November 2024

Abstract: This study conducts a comparative analysis of various machine learning and deep learning models for predicting order quantities in supply chain tiers. The models employed include XGBoost, Random Forest, CNN-BiLSTM, Linear Regression, Support Vector Regression (SVR), K-Nearest Neighbors (KNN), Multi-Layer Perceptron (MLP), Recurrent Neural Network (RNN), Bidirectional LSTM (BiLSTM), Bidirectional GRU (BiGRU), Conv1D-BiLSTM, Attention-LSTM, Transformer, and LSTM-CNN hybrid models. Experimental results show that the XGBoost, Random Forest, CNN-BiLSTM, and MLP models exhibit superior predictive performance. In particular, the XGBoost model demonstrates the best results across all performance metrics, attributed to its effective learning of complex data patterns and variable interactions. Although the KNN model also shows perfect predictions with zero error values, this indicates a need for further review of data processing procedures or model validation methods. Conversely, the BiLSTM, BiGRU, and Transformer models exhibit relatively lower performance. Models with moderate performance include Linear Regression, RNN, Conv1D-BiLSTM, Attention-LSTM, and the LSTM-CNN hybrid model, all displaying relatively higher errors and lower coefficients of determination (R^2). As a result, tree-based models (XGBoost, Random Forest) and certain deep learning models like CNN-BiLSTM are found to be effective for predicting order quantities in supply chain tiers. In contrast, RNN-based models (BiLSTM, BiGRU) and the Transformer show relatively lower predictive power. Based on these results, we suggest that tree-based models and CNN-based deep learning models should be prioritized when selecting predictive models in practical applications.

Keywords: supply chain; order prediction; deep learning; machine learning; hybrid model; supply chain tier

1. Introduction

In the contemporary global economy, the efficacy of a company's Supply Chain Management (SCM) strategy has become a crucial determinant of its competitiveness and operational efficiency (Chopra and Meindl, 2016). Accurately predicting order quantities at various tiers within the supply chain is essential for inventory management, production planning, and seamless logistics operations. However, the complexity and volatility of the supply chain environment often make accurate forecasting challenging when relying solely on traditional prediction techniques (Goodfellow et al., 2016; Ivanov and Dolgui, 2020). An increase in uncertainties at different stages of the supply chain can result in a decline in the overall efficiency of the supply network (Kouvelis et al., 2006).

Advancements in Artificial Intelligence (AI) technologies have led to the widespread adoption of deep learning and machine learning models for solving prediction problems across various industries. These models excel at learning complex data patterns and achieving high prediction accuracy based on those patterns (Ayus et al., 2023; Schmidhuber, 2015). In the realm of SCM, there is a growing trend to apply these AI-based predictive models to enhance forecasting performance (LeCun et al., 2015). Given the complexity and volatility inherent in supply chains, AI models can offer superior predictive capabilities compared to traditional methods.

The purpose of this study is to apply various deep learning, machine learning, and hybrid deep learning models to predict order quantities within supply chain tiers and to compare and analyze their performance. Through this, we aim to identify the optimal model that can maximize prediction accuracy in supply chain management. The models employed in this research include machine learning models such as XGBoost, Linear Regression, SVR, Random Forest, KNN, and MLP, as well as deep learning and hybrid models like Recurrent Neural Network (RNN), Bidirectional LSTM (BiLSTM), Bidirectional GRU (BiGRU), CNN-BiLSTM, Conv1D-BiLSTM, Attention-LSTM, Transformer, and LSTM-CNN hybrid models (LeCun et al., 2015; Schmidhuber, 2015; Vaswani et al., 2017).

Data generated within supply chains typically exhibit high dimensionality and non-linearity, adding complexity to prediction tasks (Vaswani et al., 2017). To address these challenges, this study aims to achieve high predictive performance by effectively learning complex patterns using various AI models. Deep learning models, in particular, have demonstrated exceptional performance in processing time-series data and complex data structures. Leveraging these characteristics can play a crucial role in forecasting order quantities at supply chain tiers (LeCun et al., 2015).

This research differentiates itself from previous studies in several ways. First, we conduct a comprehensive comparative analysis of various machine learning and deep learning models (e.g., XGBoost, Linear Regression, Random Forest, CNN-BiLSTM), providing detailed evaluations of each model’s predictive performance. Unlike prior studies that focused on specific models, our holistic review of diverse approaches enhances practical applicability. Second, by simultaneously analyzing the performance of tree-based models and deep learning models, we elucidate the relative strengths of both methodologies. Third, we experimentally validate the effectiveness of hybrid models (e.g., CNN-BiLSTM), demonstrating how new model architectures can be utilized to tackle supply chain forecasting challenges. In these respects, this study offers originality and practical utility that set it apart from existing research.

Furthermore, this study empirically demonstrates that the adoption of AI technologies can provide tangible benefits in supply chain management. Accurate order forecasting can contribute to reduced lead times, improved inventory management, and enhanced resilience of the entire supply chain (Schmidhuber, 2015). Consequently, companies can maximize operational efficiency in their supply chains based on improved predictive capabilities and secure a competitive advantage.

2. Literature review

Deep learning and machine learning are considered highly effective tools that can be applied to supply chain management. These technologies can learn complex data patterns, process large-scale data in real-time, and significantly improve prediction accuracy. Several studies have highlighted the importance of deep learning and machine learning technologies in supply chain management.

Cannas et al. (2024) emphasized that AI can enhance competitiveness in Operations and Supply Chain Management (OSCM) by reducing costs, shortening lead times, and improving service levels. However, they also pointed out barriers such as the need for data quality assurance, skill shortages, and high investment costs in AI adoption. Their research presents both the benefits and obstacles of AI, providing useful guidelines for future research and practice.

Culot et al. (2024) noted that AI is playing an increasingly important role in supply chain functions, confirming that technologies like machine learning, deep learning, Natural Language Processing (NLP), and computer vision are being used in data analytics, demand forecasting, and logistics optimization. The application of these technologies greatly enhances supply chain efficiency.

Ashraf et al. (2024) proposed a method for real-time detection and analysis of disruptions in cognitive digital supply chain twins using a hybrid deep learning model, focusing on enhancing supply chain resilience. This approach opens new possibilities in supply chain management.

Singh (2023) explained that AI and machine learning play crucial roles in facilitating digital transformation and securing competitive advantages in supply chain management. These technologies optimize supply chain efficiency and reduce costs, thereby improving overall resilience.

Pietukhov et al. (2023) suggested a hybrid forecasting model that integrates the lean maturity assessment of supply chains with logistic regression and neural network technologies, indicating the potential for advancement in supply chain analytics.

Kassa et al. (2023) argued that as supply chains become more vulnerable to various disruptions, AI technologies have the potential to strengthen supply chain resilience beyond what traditional risk management methods can offer.

Stranieri and Stella (2022) analyzed the performance of addressing inventory management problems using the latest Deep Reinforcement Learning (DRL) algorithms and applying them to supply chain management.

Yang et al. (2023) emphasized that in the context of the COVID-19 pandemic significantly impacting supply chain risk management, machine learning technologies can play a vital role in preventing supply chain risks and enhancing response speed.

Rolf et al. (2024) highlighted the growing importance of unsupervised learning as the complexity of supply chain management increases. Unsupervised learning is widely applied in areas like location planning and vehicle routing optimization by extracting new insights from unstructured data.

The studies mentioned above emphasize that AI and machine learning technologies play a crucial role in maximizing efficiency and resilience in supply chain management. These technologies improve supply chain performance in various aspects such as cost reduction, lead time shortening, service level improvement, real-time disruption detection, and risk prediction. However, they also point out that the successful adoption of AI and machine learning requires overcoming several obstacles, including ensuring data quality, addressing skill shortages, and managing high investment costs.

In the field of demand forecasting within supply chains, numerous studies have been conducted to efficiently apply deep learning and machine learning algorithms (Ahn et al., 2024; Douaioui et al., 2024).

Aamer et al. (2021) revealed that machine learning algorithms like neural networks, artificial neural networks, and support vector machines are widely used for demand forecasting in supply chains, particularly in the industrial sector.

Jahin et al. (2024) proposed a Multi-Channel Data Fusion Network (MCDFN) that integrates CNN, Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) to effectively extract spatial and temporal features of time-series data, explaining that this model outperforms other deep learning models.

Husna et al. (2023) conducted demand forecasting in the retail industry using machine learning and deep learning techniques to reduce costs and optimize inventory, reporting that the Random Forest algorithm showed the most efficient and highly accurate predictions.

Tirkolaee et al. (2021) investigated how machine learning technologies can be used to overcome the limitations of traditional analytical methods as the amount of data increases in supply chain management. They suggested potential contributions in various areas of supply chain management, including supplier selection, supply chain risk prediction, and production and inventory management.

Zohdi et al. (2022) compared various machine learning algorithms, including Extreme Learning Machine (ELM), to analyze the accuracy of intermittent demand forecasting, indicating that artificial neural network-based methods showed the best performance. They also explained that while deep learning and machine learning techniques are effectively applied in supply chain demand forecasting, there is no dominant algorithm across all areas of supply chain demand forecasting.

The common conclusion of these studies is that deep learning and machine learning algorithms are highly effective in demand forecasting within supply chains. These technologies are widely used across various industrial sectors and demonstrate superior performance over traditional methods, especially in processing complex time-series data and achieving prediction accuracy. Additionally, machine learning and deep learning models can make significant contributions across various areas—including supplier selection, risk prediction, production, and inventory management—in the modern supply chain environment where data volume is rapidly increasing.

Therefore, the goal of this study is to apply various deep learning and machine learning techniques to order prediction in supply chains and to experimentally compare the performance of each technique to propose the optimal predictive model. Through this, we aim to assist supply chain management practitioners in selecting effective predictive models when making data-driven decisions.

3. Model description

3.1. Supply chain model

The supply chain considered in this study consists of a Manufacturer, Distributor, Wholesaler, Retailer, and Customer (**Figure 1**).

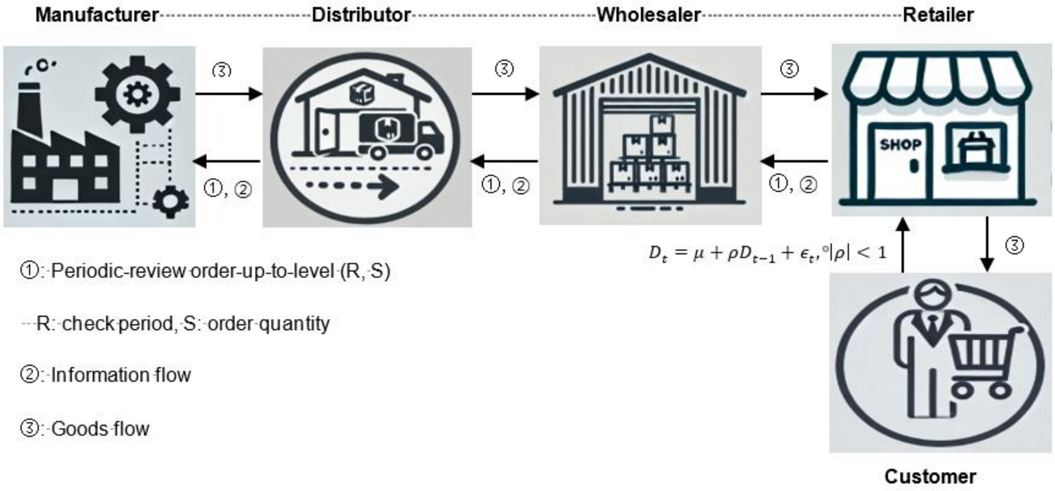

Manufacturer
Distributor
Wholesaler
Retailer
③
①, ②
SHOP
$D_t = \mu + \rho D_{t-1} + \epsilon_t, |\rho| < 1$
①: Periodic-review order-up-to-level (R, S)
R: check period, S: order quantity
②: Information flow
③: Goods flow
Customer

Figure 1. Supply chain model.

Customer demand in the supply chain is assumed to be stochastic rather than deterministic to reflect real-world conditions. It is modeled using the Autoregressive (AR) (1) process as expressed in Equation (1):

$$D_t = \mu + \rho D_{t-1} + \epsilon_t, |\rho| < 1 \quad (1)$$

In Equation (1), D_t and D_{t-1} represent customer demand at time t and t−1, respectively, while μ denotes a constant greater than or equal to zero. ϵ_t is an error term at time t that is independently and identically distributed with a mean of 0 and σ^2, and ρ represents the first-order autocorrelation coefficient. The AR (1) model explains how customer demand fluctuates over time and is useful for analyzing the trend and variability of demand across the entire supply chain.

In Equation (1), the factors affecting the customer demand process are σ and ρ. The parameter σ represents the degree of demand fluctuation and indicates the variance of demand. The parameter ρ explains the trend of customer demand and demonstrates the autocorrelation of demand.

In **Figure 1**, when the customer demand described by Equation (1) occurs, an order is placed with the Retailer tier, and this order is subsequently propagated upstream. In this process, the role of each tier is defined as ordering to the upstream tier an order quantity that is the sum of the expected demand and the order quantity received from the downstream tier. For a tier that has received an order from the downstream tier, the expected demand at time t, denoted as ED_t, is calculated as the sum of the incoming order from the downstream tier at time t−1, IO_{t-1}, and the expected demand of the tier at time t−1, ED_{t-1}, weighted by the relative weight between the order quantity and expected demand, represented by θ (where $0 \leq \theta \leq 1$). This relationship is expressed in Equation (2):

$$ED_t = \theta \cdot ED_{t-1} + (1 - \theta) \cdot IO_{t-1} \quad (2)$$

In Equation (2), when $\theta = 0$, the expected demand of the tier is not considered, and only the incoming orders from the downstream tier are taken into account. Conversely, when $\theta = 1$, the incoming orders from the downstream tier are ignored, and only the expected demand of the tier itself is considered. This model illustrates how each tier weighs and incorporates orders and demand, enabling a more precise understanding of the interactions between tiers.

Each tier employs a periodic review order-up-to-level (R, S) inventory management policy, where R represents the review period and SSS denotes the order-up-to-level. This policy involves regularly reviewing inventory at set intervals and ordering up to a predetermined level to maintain stock. Doing so enhances inventory optimization within the supply chain and increases flexibility in responding to demand fluctuations

3.2. Data collection and preprocessing

In the supply chain model illustrated in **Figure 1**, stochastic customer demand D_t follows an AR (1) process with mean (μ) set to 100, customer demand variability (σ) set to 10, and a demand trend parameter (ρ) value of −0.6. These parameters are chosen to replicate realistic demand patterns observed in supply chain simulations. Here, μ represents the average demand level, σ indicates the variability of demand, and ρ denotes the autocorrelation between demands, reflecting the persistence or reversibility of demand over time.

The simulation runs for 600 weeks (t = 1, …, 600), during which customer demand expressed by Equation (1) is delivered to the retailer, we calculate the expected demand, ED_t at each tier (Manufacturer, Distributor, Wholesaler, Retailer). To enhance robustness and ensure statistical reliability, the simulation is repeated 170 times, with each run utilizing a unique random seed. This repetition captures variability in demand, allowing us to better assess the accuracy of demand forecasting at each tier and compare model performance under diverse conditions.

Thus, the total experimental dataset comprises 408,000 samples (4 tiers × 170 sets × 600 weeks), enabling comprehensive analysis across different demand scenarios. This high volume of data increases the statistical power of the results and allows for rigorous testing of each model's robustness under varied demand patterns.

In the data preprocessing phase, missing values were addressed by imputing them with the mean of each variable's historical values, ensuring no data loss while maintaining consistency. Data normalization was also applied to rescale features to a standard range, reducing the impact of scale differences among features and improving model convergence during training.

Feature engineering techniques were utilized to enhance model interpretability and predictive power. Key features derived from the original demand data included demand lag features (previous period demands), which capture temporal patterns, and rolling mean and standard deviation features, which represent recent demand trends and volatility. Interaction terms between certain features were also generated to capture complex dependencies, which are especially beneficial for tree-based models. Data is split into 64% training data, 16% validation data, and 20% test data to ensure that an adequate amount of data is available for each stage of model development. This split minimizes the risk of overfitting, helping to ensure reliable model generalization. Training data are used to optimize model parameters, validation data for hyperparameter tuning and model selection, and test data for assessing the final model's performance on unseen data. This split allows us to confirm how effectively the model generalizes to real-world conditions and different scenarios, thus providing a realistic assessment of performance.

3.3. Applied models

To verify the accuracy of supply chain demand forecasting, this study applies various deep learning models, machine learning models, and hybrid deep learning models. Each model was selected for its suitability in processing and predicting time-series data and its ability to learn complex demand patterns that may occur in supply chains.

The deep learning models used are RNN, Bidirectional Long Short-Term Memory (BiLSTM), Bidirectional Gated Recurrent Unit (BiGRU), Attention-LSTM, and Transformer. These models are specifically designed to effectively capture the characteristics of time-series data and are expected to enhance performance in demand forecasting.

The machine learning models include XGBoost, Linear Regression, SVR, Random Forest, KNN, and MLP. These models are relatively simple, offer fast computational performance, and have the advantage of handling various data patterns.

Table 1. Hyperparameters of models.

Model	**Hyperparameters**
RNN	units = 150 (1st RNN layer), units = 100 (2nd RNN layer), dropout = 0.2
BiLSTM	units = 150 (1st LSTM layer), units = 100 (2nd LSTM layer), dropout = 0.2
BiGRU	units = 150 (1st GRU layer), units = 100 (2nd GRU layer), dropout = 0.2
Attention-LSTM	units = 100 (LSTM), num_heads = 2, key_dim = 32 (Attention)
Transformer	num_heads = 4, key_dim = 32 (MultiHeadAttention), units = 100 (Dense)
XGBoost	• n_estimators: 100 (number of trees to use) • learning_rate: 0.1 (learning rate) • max_depth: 6 (maximum depth of the trees) • min_child_weight: 1 (minimum sum of weights of all observations required in a child node) • subsample: 1.0 (ratio of samples to use for training) • colsample_bytree: 1.0 (ratio of features to use per tree) • gamma: 0 (minimum loss reduction required to make a further partition on a leaf node) • reg_alpha (alpha): 0 (weight of the L1 regularization term) • reg_lambda (lambda): 1 (weight of the L2 regularization term) • objective: ‘reg’ (loss function for regression problems)
Linear Regression	• fit_intercept: True (whether to calculate the intercept for this model) • normalize: False (whether to normalize the regressors; applicable only when fit_intercept = False) • copy_X: True (whether to copy the input data; if False, it may overwrite the input data) • n_jobs: None (whether to use all CPU cores; None means 1 unless in a joblib.parallel_backend context)
SVR (Linear)	kernel = ‘linear’
Random Forest	n_estimators = 100, random_state = 42
K-Nearest Neighbors	n_neighbors = 5
MLP (Multi-Layer Perceptron)	hidden_layer_sizes = (100,100), max_iter = 500, random_state = 42
CNN-BiLSTM	filters = 64, kernel_size = 3 (Conv1D), units = 100 (LSTM), dropout = 0.2
Conv1D-BiLSTM	filters = 64, kernel_size = 3 (Conv1D), units = 100 (LSTM), dropout = 0.2
LSTM-CNN	units = 100 (LSTM), filters = 64, kernel_size = 3 (Conv1D)

The hybrid deep learning models are CNN-BiLSTM, Conv1D-BiLSTM, and LSTM-CNN. These models combine the feature extraction capabilities of CNN with the time-series data learning abilities of LSTM to maximize predictive performance.

The hyperparameters applied for each model are summarized in **Table 1**.

When setting the hyperparameters for each model, this study followed configurations suggested to yield optimal performance for time-series models like RNN, LSTM, and GRU, as presented in the studies by Greff et al. (2017) and Jozefowicz et al. (2015). Specifically, we adopted the units and dropout values recommended in their research. The hyperparameters for XGBoost were based on the default settings proposed by Chen and Guestrin (Chen and Guestrin, 2016) and Prokhorenkova et al. (2018), where parameters such as n_estimators, learning_rate, and max_depth have been reported to significantly influence performance.

For the Transformer model, we adhered to the architecture and hyperparameter settings suggested by Vaswani et al. (2017). Their study explained that hyperparameters like num_heads and key_dim in the Multi-Head Attention structure critically affect model performance. In alignment with this, we set the values of num_heads and key_dim to optimize the Transformer's performance in our study.

The stopping conditions for each model are summarized in **Table 2**. All deep learning models applied common stopping conditions using the EarlyStopping callback, with monitor = 'val_loss', patience = 10, and restore_best_weights = True.

Table 2. Stopping conditions of models.

Model	Stopping conditions
RNN	Train for a maximum of 200 epochs or terminate early if EarlyStopping conditions are met.
BiLSTM	Train for a maximum of 200 epochs or terminate early if EarlyStopping conditions are met.
BiGRU	Train for a maximum of 200 epochs or terminate early if EarlyStopping conditions are met.
Attention-LSTM	Train for a maximum of 200 epochs or terminate early if EarlyStopping conditions are met.
Transformer	Train for a maximum of 200 epochs or terminate early if EarlyStopping conditions are met.
XGBoost	Training concludes after all n_estimators (100 trees by default) are generated- Since early_stopping_rounds is not set, all trees are trained
Linear Regression	No explicitly defined stopping condition; training concludes when optimal weights are learned from the data.
SVR (Linear)	Training stops after reaching the default max_iter (maximum iterations, default is -1 for unlimited iterations) or when convergence criteria are met.
Random Forest	The training concludes after all n_estimators (100 trees by default) are generated.
K-Nearest Neighbors	
MLP (Multi-Layer Perceptron)	Training stops after reaching max_iter (maximum iterations, default is 200) or when convergence criteria are met.
CNN-BiLSTM	Train for a maximum of 200 epochs or terminate early if EarlyStopping conditions are met.
Conv1D-BiLSTM	Train for a maximum of 200 epochs or terminate early if EarlyStopping conditions are met.
LSTM-CNN	Train for a maximum of 200 epochs or terminate early if EarlyStopping conditions are met.

As demonstrated, most machine learning models terminate based on predefined iteration counts or convergence criteria. In contrast, deep learning models employ EarlyStopping callbacks to halt training early depending on improvements in validation loss. This approach helps prevent overfitting and enhances training efficiency.

4. Results and analysis

In this study, we compared the performance of various machine learning and deep learning models by taking into account factors such as customer demand, expected demand, and hyperparameters in the supply chain model. The predictive performance of each model was evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2). **Table 3** presents the experimental results of each model.

Table 3. Experimental results of models.

Model	MSE	MAE	RMSE	R^2
RNN	1.688564	1.076484	1.299448	−0.366993
BiLSTM	3.463235	1.492647	1.860977	−1.803694
BiGRU	3.303734	1.463424	1.817618	−1.674569
Attention-LSTM	9.171739×10^{-1}	0.782845	0.957692	0.257493
Transformer	3.421401	1.485400	1.849703	−1.769826
XGBoost	3.727327×10^{-9}	0.000050	0.000061	1.000000
Linear Regression	1.153874	0.676716	1.074185	0.065871
SVR (Linear)	7.415028×10^{-2}	0.197907	0.272305	0.939971
Random Forest	1.720588×10^{-4}	0.001324	0.013117	0.999861
K-Nearest Neighbors	0.000000	0.000000	0.000000	1.000000
MLP (Multi-Layer Perceptron)	3.988533×10^{-3}	0.047603	0.063155	0.996771
CNN-BiLSTM	7.823669×10^{-4}	0.020056	0.027971	0.999367
Conv1D-BiLSTM	6.693360×10^{-1}	0.682174	0.818130	0.458133
LSTM-CNN	1.147804	0.914008	1.071356	0.070784

The XGBoost model exhibited the best predictive performance, with MSE, MAE, and RMSE values approaching zero and an R^2 value of 1.0. This suggests that XGBoost effectively learned complex patterns and interactions among variables in the data.

The Random Forest model also demonstrated very high performance, with an MSE of 1.720588×10^{-4}, MAE of 0.001324, RMSE of 0.013117, and an R^2 of 0.999861. This result can be attributed to the ensemble technique used by Random Forest, which combines multiple decision trees to enhance predictive power.

The CNN-BiLSTM model showed the best performance among deep learning models, with an MSE of 7.823669×10^{-4}, MAE of 0.020056, RMSE of 0.027971, and an R^2 of 0.999367. This high prediction accuracy was achieved by combining the feature extraction capabilities of CNN with the time-series learning abilities of BiLSTM.

The MLP model also exhibited excellent performance, with an MSE of 3.988533×10^{-3}, MAE of 0.047603, RMSE of 0.063155, and an R^2 of 0.996771.

In contrast, the BiLSTM, BiGRU, and Transformer models showed relatively poor performance, with high MSE and MAE values and negative R^2 values. This indicates that these models did not effectively learn from the dataset used in this study.

The Linear Regression model demonstrated moderate performance, with an MSE of 1.153874, MAE of 0.676716, RMSE of 1.074185, and R^2 of 0.065871. This suggests that the assumption of linearity in the data may not have been sufficiently met.

The KNN model showed MSE, MAE, and RMSE values all equal to zero and an R^2 value of 1.0. However, this is an unrealistic result, raising suspicions of data leakage or errors in the model evaluation method. Therefore, the results of the KNN model are unreliable and require additional review.

The SVR (Linear) model exhibited relatively high performance with an R^2 value of 0.939971, but its MSE and MAE values were higher compared to the tree-based models.

The Conv1D-BiLSTM, Attention-LSTM, and LSTM-CNN hybrid models showed moderate performance, with relatively high MSE and MAE values and low R^2 values.

Figure 2 illustrates a bar graph comparing the MSE, MAE, RMSE, and R^2 values of each model. This confirms that models like XGBoost, Random Forest, and CNN-BiLSTM exhibited superior performance. In contrast, BiLSTM, BiGRU, and Transformer models showed relatively lower performance.

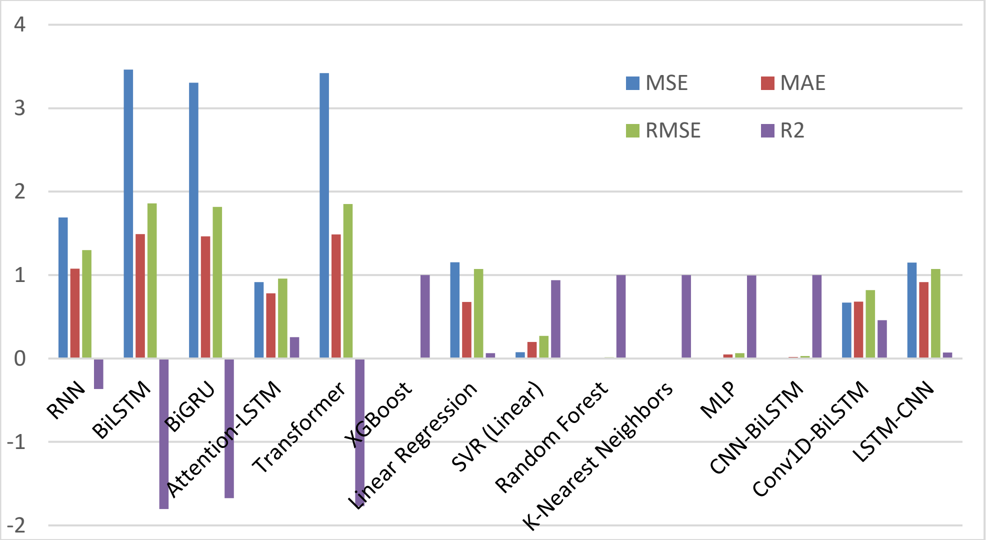


Figure 2. Comparison of model performance metrics (MSE, MAE, RMSE, R^2).

To evaluate the variability and stability of the predictive performance of each model, we visualized the distribution of MSE and MAE for each model using box plots in **Figure 3**.

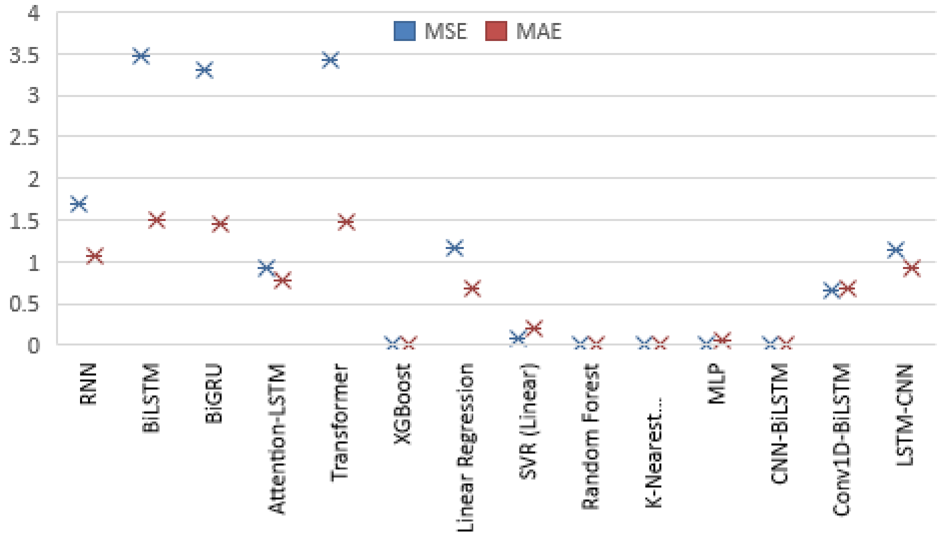


Figure 3. Distribution of MSE and MAE for each model.

Since MSE is sensitive to large errors, the MSE value significantly increases when large errors occur. From the MSE distributions of each model in **Figure 3**, the following interpretations can be made:

XGBoost, Random Forest, and K-Nearest Neighbors models have MSE values close to zero, with very narrow box plot ranges and lower and upper bounds at almost the same position. This suggests that these models have very little prediction error and highly stable predictive performance. Particularly, XGBoost and Random Forest, as tree-based ensemble models, effectively learn complex patterns in the data, resulting in outstanding predictive performance.

CNN-BiLSTM and MLP models show very low MSE values with relatively narrow distributions. This indicates that deep learning–based models perform well on supply chain data and have relatively small prediction errors. Specifically, the CNN-BiLSTM model exhibits high performance by combining the feature extraction capabilities of CNN with the time-series learning abilities of BiLSTM.

BiLSTM, BiGRU, and Transformer models show very high MSE values with wide distributions. This indicates an unstable predictive performance and a tendency to record large errors in any data. In particular, the Transformer model may have been overly sensitive to specific characteristics of the data or failed to learn effectively.

In **Figure 3**, MAE represents the mean of the absolute values of the errors, assigning equal importance to all errors. The MAE distributions of each model exhibit the following characteristics:

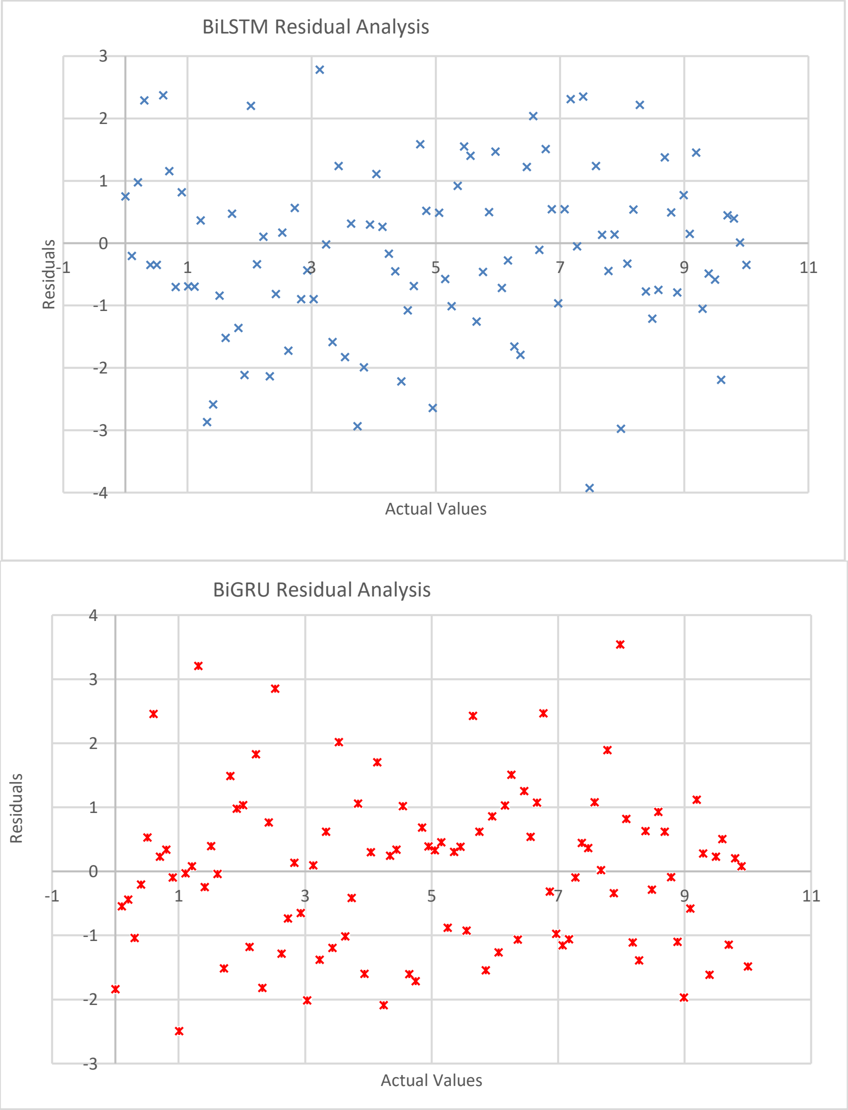
XGBoost, Random Forest, and KNN models have very low MAE values with narrow box plot ranges. This suggests that the absolute prediction errors are very small and that the models showed consistent performance across the entire dataset. Notably, the KNN model shows zero errors for all predictions, which may indicate overfitting or peculiarities in data processing, necessitating further investigation in future research.

CNN-BiLSTM and MLP models display low MAE values with narrow distributions, indicating small prediction errors and consistent performance. This reaffirms that deep learning–based models have high predictive accuracy.

BiLSTM, BiGRU, and Transformer models have relatively high MAE values with wide distributions. This suggests that these models exhibit large prediction errors on specific data points and have unstable performance. Despite having bidirectional capabilities in time-series learning, BiLSTM and BiGRU show large prediction errors on this dataset.

Linear Regression, RNN, Conv1D-BiLSTM, and Attention-LSTM models have MAE distributions that are moderately dispersed, indicating that while they perform at a certain level, they do not achieve optimal performance.

Next, we performed residual analysis on the models (BiLSTM, BiGRU, and Transformer) with low predicted performance to evaluate how these models learned in the supply chain demand forecasting problem. **Figure 4** shows scatter plots illustrating the relationship between the residuals and actual values for each model.



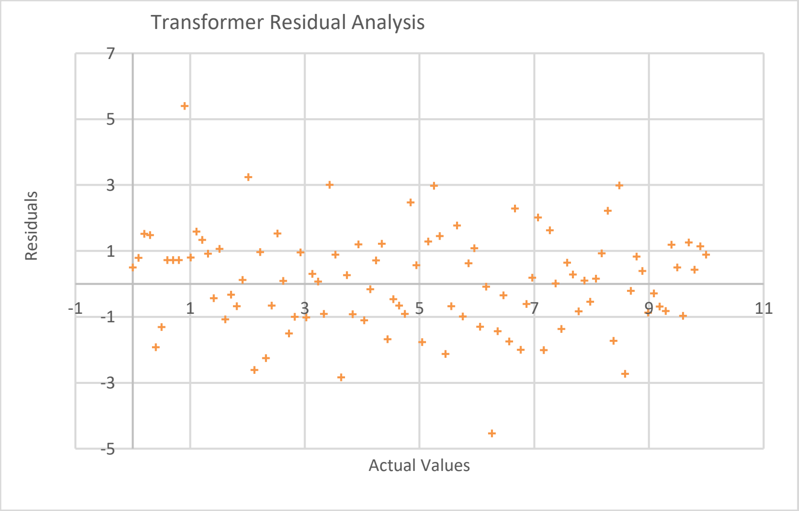


Figure 4. Residual Analysis of BiLSTM, BiGRU, and transformer models.

The residual analysis of the BiLSTM model is shown in the first graph of **Figure 4**. Although the BiLSTM model has the characteristic of being able to learn time-series data bidirectionally, it exhibited large prediction errors in this experiment. As seen in the scatter plot, many data points have residuals that deviate significantly from the baseline zero. Particularly, the residuals are not evenly distributed and tend to be large in specific intervals. These results suggest that the BiLSTM model did not effectively learn the patterns in the supply chain data.

The residual analysis of the BiGRU model is shown in the second graph of **Figure 4**. Similar to BiLSTM, the BiGRU model can learn time-series data bidirectionally and has higher computational efficiency due to its GRU structure. However, in this experiment, the BiGRU model also showed large prediction errors. As observed in the scatter plot, the residuals are not concentrated around the baseline zero but are widely dispersed.

The residual analysis of the Transformer model is presented in the third graph of **Figure 4**. The Transformer model is based on the attention mechanism and has various applications in processing time-series data. However, in the supply chain data of this study, the Transformer model showed poor predictive performance. The residual analysis reveals that many points have residuals significantly deviating from the baseline zero. Particularly, the residuals are asymmetrically distributed, with extreme errors in certain regions. This indicates that the Transformer model was not effectively trained on the dataset and may have reacted overly sensitive to specific time-series patterns or noise in the data.

In summary, we confirmed that tree-based models (XGBoost, Random Forest) and the CNN-BiLSTM model are most effective for predicting supply chain order quantities. In contrast, RNN-based models (BiLSTM, BiGRU) and the Transformer model showed relatively poor performance. These results suggest that data characteristics and model suitability must be considered when selecting predictive models.

5. Discussions and conclusions

This study conducted a comparative analysis of the predictive performance of diverse machine learning and deep learning models on a supply chain order dataset. The performance of each model was evaluated using metrics such as MSE, MAE, RMSE, and the coefficient of determination (R^2).

Tree-based models, particularly XGBoost and Random Forest, along with the CNN-BiLSTM model, demonstrated the superior performance of the supply chain model. Notably, the XGBoost model achieved the highest accuracy across all evaluation metrics, including an R^2 value of 1.0, which reflects the model's strong capacity to capture complex patterns and feature interactions. This result suggests that the XGBoost model, by utilizing its ability to learn intricate relationships in the data, is highly suited for supply chain predictions. Similarly, the Random Forest model performed effectively, supporting its robustness in leveraging multiple decision trees to capture various aspects of the data, thus providing high predictive accuracy. The CNN-BiLSTM and MLP models also performed well, showcasing the potential of deep learning approaches in identifying complex patterns in supply chain data.

However, RNN-based models, such as BiLSTM, BiGRU, and the Transformer model, exhibited relatively poor performance, characterized by high error values and low or negative R^2 values. These results indicate that, although RNN-based models generally excel in time-series analysis, they may face challenges when applied to datasets with different characteristics, such as the supply chain data used in this study. The Transformer model showed a similar performance limitation, reinforcing the possibility that these models are less suited for capturing complex static interactions within non-sequential datasets like this one.

Models with moderate performance, including Linear Regression, RNN, Conv1D-BiLSTM, Attention-LSTM, and the LSTM-CNN hybrid model, exhibited higher error metrics and lower R^2 values than top-performing models. These results suggest that such models may not fully explain the variance in the supply chain dataset or might exhibit poor data fit, limiting their applicability in this context.

The findings of this study provide practical insights into the use of predictive models in supply chain management. Tree-based models, due to their high accuracy and robustness, are recommended for applications requiring precise order quantity predictions. The CNN-BiLSTM model also shows promise, especially when more complex pattern extraction is necessary. Conversely, RNN-based models and the Transformer model may require modifications or enhanced data preprocessing to improve their applicability in this domain.

It is important to note that the study has limitations, primarily in its model-specific approach to supply chain data. Future research may consider incorporating external factors such as economic indicators or weather data to assess whether these models can improve their performance with additional contextual information. Additionally, the study did not extensively evaluate the computational requirements of each model, which is crucial for assessing practical feasibility in real-time applications. Including such analysis in future studies would allow for a better understanding of each model's resource efficiency, particularly for business implementations. Building on this study's findings, future research can aim to optimize model performance through various techniques:

1) Enhanced Feature Engineering: Conducting feature importance analysis and refining input variables could help identify and exclude less impactful variables. Additionally, advanced feature engineering, including the creation of new features, may further improve model accuracy.
2) Hyperparameter Optimization: Using techniques like grid search or Bayesian optimization could enhance the performance of models, especially tree-based models like XGBoost and Random Forest, which are highly sensitive to parameter tuning.
3) Cross-Validation: Implementing k-fold cross-validation would provide a more robust evaluation of the model's generalizability and help reduce overfitting risks.
4) Ensemble and Regularization Techniques: Applying ensemble methods, such as stacking or bagging, along with advanced Transformer architectures, could further refine predictions. Regularization techniques, including dropout, would also aid in improving model generalization.
5) Computational Complexity and Resource Analysis: Examining the computational load of each model could provide practical insights into model efficiency, helping to determine which models are suitable for deployment in real business environments.

Lastly, future studies should also focus on analyzing the bias-variance trade-off to evaluate each model's tendency to overfit or underfit, which is crucial for determining model suitability in supply chain prediction.

In conclusion, this study evaluated the performance of various models for supply chain order predictions, with XGBoost, Random Forest, and the CNN-BiLSTM model emerging as the most effective. In contrast, RNN-based models and the Transformer model demonstrated lower accuracy. These findings suggest prioritizing tree-based models and CNN-based deep learning models for practical applications, while future studies may explore additional strategies to further enhance model performance and usability.

Acknowledgments: This research was conducted by research funds from Gwangju University in 2024.

Conflict of interest: The author declares no conflict of interest.

References

Aamer, A. M., Yani, L. P. E., & Priyatna, I. M. A. (2021). Data analytics in the supply chain management: Review of machine learning applications in demand forecasting. Operations and Supply Chain Management, 14(1), 1–13. http://doi.org/10.31387/oscm0440281.

Ahn, H. I., Song, Y. C., Olivar, S., Mehta, H., & Tewari, N. (2024). GNN-based probabilistic supply and inventory predictions in supply chain networks. arXiv preprint arXiv:2404.07523.

Ashraf, M., Eltawil, A., & Ali, I. (2024). Disruption detection for a cognitive digital supply chain twin using hybrid deep learning. Operational Research, 24, 23. https://doi.org/10.48550/arXiv.2309.14557.

Ayus, I., Natarajan, N., & Gupta, D. (2023). Comparison of machine learning and deep learning techniques for the prediction of air pollution: A case study from China. Asian Journal of Atmospheric Environment, 17(4), 1–22. https://doi.org/10.1007/s44273-023-00005-w.

Cannas, V. G., Ciano, M. P., Saltalamacchia, M., & Secchi, R. (2024). Artificial intelligence in supply chain and operations management: a multiple case study research. International Journal of Production Research, 62(9), 3333–3360. DOI:10.1080/00207543.2023.2232050.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,785–794. https://dl.acm.org/doi/pdf/10.1145/2939672.2939785.

Chopra, S., & Meindl, P. (2016). Supply Chain Management: Strategy, Planning, and Operation. Pearson Education.

Culot, G., Podrecca, M., & Nassimbeni, G. (2024). Artificial intelligence in supply chain management: A systematic literature review of empirical studies and research directions. Computers in Industry, 162, 103813. DOI:10.1016/j.compind.2024.104132.

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. 7(5):93. https://doi.org/10.3390/asi7050093.

Foumani, N. M., Miller, L., Tan, C. W., Webb, G. I., Forestier, G., & Salehi, M. (2024). Deep learning for time series classification and extrinsic regression: A current survey. ACM Computing Surveys, 56(9), Article 217. https://doi.org/10.48550/arXiv.2302.02515.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems, 28(10), 2222–2232. https://doi.org/10.48550/arXiv.1503.04069.

Husna, A. U., Amin, S. H., & Ghasempoor, A. (2023). Demand forecasting using machine learning and deep learning approaches in the retail industry: A comparative study. In Industrial Engineering in the Covid-19 Era (249–264). Springer, Cham. https://doi.org/10.1007/978-3-031-25847-3_24.

Ivanov, D., & Dolgui, A. (2020). Viability of intertwined supply networks: Extending the supply chain resilience angles towards survivability. International Journal of Production Research, 58(10), 2904–2915. https://doi.org/10.1080/00207543.2020.1750727.

Jahin, M. A., Shahriar, A., & Al Amin, M. (2024). MCDFN: Supply chain demand forecasting via an explainable multi-channel data fusion network model. arXiv, 2405.15598. https://doi.org/10.48550/arXiv.2405.15598.

Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In Proceedings of the 32nd International Conference on Machine Learning (2342–2350). https://dl.acm.org/doi/10.5555/3045118.3045367.

Kassa, A., Kitaw, D., Stache, U., Beshah, B., & Degefu, G. (2023). Artificial intelligence techniques for enhancing supply chain resilience: A systematic literature review, holistic framework, and future research. Computers & Industrial Engineering, 186, 109714. DOI:10.1016/j.cie.2023.109714.

Kouvelis, P., Chambers, C., & Wang, H. (2006). Supply chain management research and production and operations management: review, trends, and opportunities. Production and Operations Management, 15(3), 449–469. https://doi.org/10.1111/j.1937-5956.2006.tb00257.x.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. DOI:10.1038/nature14539.

Lourenço, V. M., Ogutu, J. O., Rodrigues, R. A., Posekany, A., & Piepho, H. P. (2024). Genomic prediction using machine learning: A comparison of the performance of regularized regression, ensemble, instance-based and deep learning methods on synthetic and empirical data. BMC Genomics, 25(1), 152. https://doi.org/10.1186/s12864-023-09933-x.

Pietukhov, R., Ahtamad, M., Faraji-Niri, M., & El-Said, T. (2023). A hybrid forecasting model with logistic regression and neural networks for improving key performance indicators in supply chains. Supply Chain Analytics, 4, 1–9. https://doi.org/10.1016/j.sca.2023.100041.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: Unbiased boosting with categorical features. In Advances in Neural Information Processing Systems, 31. https://doi.org/10.48550/arXiv.1706.09516.

Rolf, B., Beier, A., Jackson, I., Müller, M., Reggelin, T., Stuckenschmidt, H., & Lang, S. (2024). A review on unsupervised learning algorithms and applications in supply chain management. International Journal of Production Research, 62(15), 1–51. https://doi.org/10.1080/00207543.2024.2390968.

Sangeetha, J. M., & Alfia, K. J. (2024). Financial stock market forecast using evaluated linear regression based machine learning technique. Measurement: Sensors, 31, 100950. https://doi.org/10.1016/j.measen.2023.100950.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85–117. https://doi.org/10.48550/arXiv.1404.7828.

Singh, P. K. (2023). Digital transformation in supply chain management: Artificial Intelligence (AI) and Machine Learning (ML) as catalysts for value creation. International Journal of Supply Chain Management, 12(6), 57–63. DOI:10.59160/ijscm.v12i6.6216.

Stranieri, F., & Stella, F. (2022). Comparing deep reinforcement learning algorithms in two-echelon supply chains. arXiv preprint, arXiv:2204.00000. https://doi.org/10.48550/arXiv.2204.09603.

Tirkolaee, E. B., Sadeghi Darvazeh, S., Mansoori Mooseloo, F., & Rezaei Vandchali, H. (2021). Application of machine learning in supply chain management: A comprehensive overview of the main areas. Mathematical Problems in Engineering, 2021, 1–14. DOI:10.1155/2021/1476043.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (5998–6008). http://arxiv.org/abs/1706.03762.

Yang, M., Lim, M. K., Qu, Y., Ni, D., & Xiao, Z. (2023). Supply chain risk management with machine learning technology: A literature review and future research directions. Computers & Industrial Engineering, 175, 108859. DOI: 10.1016/j.cie.2022.108859.

Zohdi, M., Rafiee, M., Kayvanfar, V., & Salamiraad, A. (2022). Demand forecasting based machine learning algorithms on customer information: An applied approach. International Journal of Information Technology, 14(3), 1–12. DOI:10.1007/s41870-022-00875-3.