

Application of the LSTM Model for Demand Forecasting in Plastic Packaging Production

Merve YILMAZ^{1*}, Alptekin DURMUŞOĞLU²

Keywords

Plastic packaging,
Demand Forecasting,
LSTM

Abstract – In the face of intensifying competition and rapidly evolving global conditions, companies are increasingly compelled to develop forward-looking strategies that support long-term sustainability. A central element of these strategies is the capability to produce reliable short- and long-term demand forecasts. Accurate forecasting not only reduces the likelihood of future revenue losses but also strengthens decision-making by enabling more effective planning and cost optimization. With dependable forecasts, managers can design strategic initiatives for upcoming periods and allocate organizational resources with greater precision. The plastic packaging sector is characterized by high dynamism and pronounced sensitivity to demand fluctuations arising from factors such as strong market rivalry, evolving consumer preferences, and variability in raw material costs. Consequently, forecasting accuracy becomes essential for maintaining efficient and uninterrupted production planning. This study applies the Long Short-Term Memory (LSTM) algorithm—an advanced machine learning technique grounded in time series modeling—to predict demand within the plastic packaging industry. Sales data recorded at 15-day intervals from January 2015 to December 2024 were analyzed, yielding a dataset of 240 observations. Seventy percent of the data was used to train the model, while the remaining thirty percent was reserved for testing. To assess the model’s predictive performance, four commonly used error-based metrics were employed: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). Metrics such as Symmetric MAPE (SMAPE) and the Coefficient of Determination (R^2) were intentionally excluded due to limitations related to interpretability and reliability in non-linear time series applications. The trained LSTM model was subsequently used to estimate monthly sales volumes for 2025. The findings provide valuable insights for enhancing production planning, reducing inventory-related expenses, and improving customer satisfaction within the plastic packaging sector, demonstrating the practical value of LSTM-based forecasting.

^{1*}Corresponding Author. Gaziantep University, Graduate School of Natural and Applied Sciences, Gaziantep, Turkey. E-mail: merve.yilmaz27.my@gmail.com  ORCID: 0009-0004-3514-0043

²Samsun University, Industrial Engineering Department, 55420 Samsun, Turkey. E-mail: alptekin.durmusoglu@samsun.edu.tr  ORCID: 0000-0001-9800-5747
Citation: Yılmaz, M. and Durmuşoğlu, A.. (2025). Application of LSTM Model for Demand Forecasting in Plastic Packaging Production. *Natural Sciences and Engineering Bulletin*, 2(2), 138-155.

1. Introduction

LSTM capable of identifying temporal patterns in historical data and adapting to sudden shifts in demand is essential. Time series forecasting methods have become increasingly important, enabling companies to estimate future order volumes, align production planning, manage inventory levels, reduce operational costs, and improve customer satisfaction. Especially in sectors closely linked to FMCG markets, precise forecasting is vital for maintaining operational agility and minimizing the risks of overproduction or stock shortages.

Although traditional forecasting methods rely on historical trends, their accuracy often deteriorates in highly dynamic environments. As a result, machine learning approaches have emerged as more effective alternatives. Among them, LSTM networks, a variant of recurrent neural networks (RNNs) have demonstrated strong capabilities in modeling complex, non-linear, time-dependent data, making them particularly suitable for demand forecasting in volatile industries such as plastic packaging.

In the plastic packaging industry, there is growing interest in time series forecasting methods that can optimize production processes, improve inventory control, and support strategic planning. Accurate forecasts derived from historical data offer companies a significant competitive edge by enhancing decision-making. Additionally, time series analysis helps determine optimal stock levels by analyzing past inventory movements, thereby reducing the risks of both overstocking and stock shortages. Given the sector's close integration with FMCG markets, effective inventory management is essential, and time series forecasting plays a key role in achieving it.

Furthermore, time series forecasting enhances firms' competitiveness and long-term strategic planning by offering a clearer understanding of current trends and likely future scenarios. In industries characterized by high uncertainty and rapid change, such as plastic packaging, the predictive capabilities of such methods are critical for attaining both short-term efficiency and long-term growth.

This study investigates the application of LSTM-based demand forecasting in the plastic packaging industry, aiming to improve prediction accuracy and strengthen operational planning. Following a brief literature review on the use of LSTM in forecasting, a model was developed using real sales data from a plastic packaging company based in Gaziantep, Turkey. The model's performance was evaluated using standard error metrics, including MSE, RMSE, MAE, and MAPE.

To assess the robustness and reliability of the LSTM model, its forecasting performance was compared with that of other traditional and advanced time series models, including Autoregressive Integrated Moving Average (ARIMA), Seasonal ARIMA (SARIMA), and Gated Recurrent Unit (GRU) networks. Among all models evaluated, the LSTM model produced the most accurate results, demonstrating superior performance in capturing complex temporal dependencies and minimizing forecast errors.

Based on these findings, monthly demand forecasts for the year 2025 were generated. The primary aim of this study is to demonstrate that LSTM-based forecasting can effectively reduce demand uncertainty, optimize production planning, and enhance customer satisfaction. In addition to its practical implications, the study contributes to the academic literature and provides actionable insights for industry decision-makers seeking data-driven solutions for complex demand forecasting challenges.

2. Literature Review

Eraslan et al. (2007) highlighted the critical role of effective hyperparameter tuning in enhancing the predictive accuracy of LSTM models for demand forecasting in the plastic packaging industry. Later, Bachu (2019) evaluated deep learning models, including LSTM, for predicting packaging demand in the logistics sector and confirmed their superior accuracy compared to conventional methods (Gür & Eşidir, 2023; Işık, 2023). Similarly, Erdal (2020) demonstrated that LSTM could accurately forecast seasonal demand fluctuations by leveraging historical sales data to detect trends and cyclical patterns.

Abbasimehr and Paki (2021) aimed to improve forecasting accuracy in time series analysis by integrating LSTM networks with attention mechanisms. While LSTM effectively captures long-term dependencies in sequential data, the attention mechanism enables the model to focus on the most relevant portions of the input sequence, thereby enhancing predictive performance. Their hybrid model outperformed traditional forecasting techniques.

Aytaç et al. (2022) compared ARIMA and LSTM models for revenue forecasting in the online gaming industry, finding that LSTM consistently outperformed ARIMA, particularly in complex and dynamic data environments. Albeladi et al. (2023) also conducted a comparative study of ARIMA and LSTM, showing that LSTM performs better on large and complex datasets, while ARIMA is more suitable for simpler time series, emphasizing the importance of selecting forecasting models according to data characteristics.

Işık (2023) noted that LSTM represents a significant advancement over traditional linear models due to its ability to learn complex, non-linear dependencies. LSTM's advanced architecture enables a deeper understanding of demand trends, facilitating optimization in production and supply chain operations (Kalinggo & Zulkarnain, 2020). Dinata and Suharjito (2024) further emphasized that the LSTM architecture, consisting of forget, input, and output gates, allows the model to retain or discard information as needed, improving efficiency by filtering out irrelevant historical data.

Akçay and Yiltas-Kaplan (2024) compared various time series forecasting models, including ARIMA and LSTM, using airtime consumption data. Their results indicated that LSTM significantly outperformed ARIMA in terms of prediction accuracy. Considering the inherently volatile nature of the plastic packaging industry, advanced forecasting models such as LSTM can enhance both competitiveness and operational sustainability, suggesting broader applications of time series forecasting techniques across industries.

Anwar et al. (2024) conducted a systematic review of the challenges and opportunities in multilayer plastic waste management, emphasizing the complexity of recycling and the need for more effective strategies. Given LSTM's ability to model complex, non-linear time series data, it is particularly suitable for representing dynamic industrial processes in sectors such as plastic packaging. Fatima and Rahimi (2024) further noted that LSTM's capacity to capture non-linear and long-term relationships enables highly accurate forecasts, supporting operational efficiency and strategic decision-making.

Çalık (2025) investigated LSTM-based anomaly detection within the IP Multimedia Subsystem (IMS), a critical telecommunication infrastructure component. The study compared LSTM with various unsupervised learning methods, including Graph-Based Detection, DBSCAN, Local Outlier Factor (LOF), Z-Score, and Isolation Forest. Results showed that LSTM outperformed these models in identifying complex, time-dependent anomalies. Its gated architecture enables retention of temporal information over long sequences, allowing better handling of noise and variability. LSTM also detected critical anomalies earlier than traditional monitoring systems. Comparisons with Convolutional Neural Networks (CNNs) indicated that LSTM generally achieves lower error rates and more efficient training, while hybrid models such as TadGAN, integrating LSTM with Generative Adversarial Networks (GANs), achieved high F1 scores in anomaly detection tasks. Nevertheless, challenges such as extended training times and high computational costs remain, motivating further research into more efficient hybrid approaches.

In the environmental domain, Tüler (2025) applied LSTM to multivariate time series data to predict both flow rate and water quality indices (WQI). The study demonstrated that LSTM outperformed traditional models, including ARIMA, support vector regression, and feedforward neural networks, in both accuracy and robustness. LSTM's strength lies in its capacity to model long-term temporal dependencies, which are critical for environmental forecasting. The study addressed challenges related to noisy and incomplete datasets through preprocessing and normalization techniques and emphasized hyperparameter tuning for optimal performance. Further research in river flow prediction, groundwater level forecasting, and pollutant estimation confirms LSTM's effectiveness in handling complex, dynamic environmental data. Despite these advancements, challenges remain in deploying LSTM models for real-time environmental monitoring, particularly regarding

data availability and computational demands. Ongoing research into hybrid models combining LSTM with other algorithms aims to improve both prediction accuracy and computational efficiency.

3. LSTM: Long Short-Term Memory Networks

LSTM is an efficient deep learning methodology for modeling long-term dependencies and complex relations in time series data. First proposed by Hochreiter and Schmidhuber in 1997, LSTM was designed to handle the "gradient loss" problem of traditional artificial neural networks and RNNs. Due to this feature, LSTM networks perform better than others in predicting future trends based on past data or analyzing sequential data. The most salient feature of LSTM is that it can learn the long-term dependencies effectively, and provide a structure suitable for modeling these dependencies. While the traditional RNNs are quite successful in capturing the short-term dependencies, they are not good enough to understand the long-term patterns in time series. This problem is solved in LSTM through cell states and gate mechanisms. In this framework, the study of Albeladi et al. (2023) conducts time series prediction using LSTM and ARIMA models. In the work, the prediction accuracy of the two models was compared.

LSTM model is a distinctive artificial neural network architecture, primarily recognized for its ability to learn long-term dependencies in time series analysis and sequential data processing. The operational principles of LSTM are based on the interaction between cell states and gate mechanisms. This architecture allows LSTM to process both short-term and long-term information simultaneously. Initially developed to address the limitations of traditional RNNs in capturing long-term dependencies, LSTM has effectively bridged this gap, particularly through its capacity to mitigate the vanishing gradient problem. The core functioning of LSTM is driven by the cell state and gate mechanisms, which regulate the flow of information within the network (Kalinggo and Zulkarnain, 2020). The cell state functions as a form of memory that carries information across time steps, enabling the model to store and retain past information without losing it during the learning process. This memory is then accessed when needed, contributing to the model's ability to make informed predictions.

The LSTM model structure uses cell states that are especially designed to learn long-term dependencies. The cell state, acting as the highway carrying past information in a time series, is modified and accessed through gate mechanisms responsible for controlling how the model accesses this information. The gate mechanisms within the LSTM model control this memory, determining which information should be retained and which should be discarded. Specifically, the input gate processes new information, such as economic indicators, which may influence future demand. This information is then integrated into the model's learning process. Additionally, the output gate determines which portions of the stored memory are relevant for use in subsequent time steps. During this process, the cell state and input data pass through a sigmoid activation function, and the resulting information is scaled using the tanh activation function. In the context of demand forecasting within the plastic packaging sector, the output gate analyzes past customer behaviors and incorporates this historical data into the forecasting process, thereby improving the accuracy of future demand predictions (Keskin and Koçoğlu, 2022).

It looks at the current input data and previous cell states to determine how much new information should be added to the cell. The input gate generally works with the activation functions sigmoid and tanh, allowing for effective processing of information into the cell state. According to Gür & Eşidir (2023), the output gate determines how much of the information in the cell state will be used in the next step. This gate takes responsibility for transferring information in the cell state to the next layer in the network. This is the structure of LSTM models that allow them to learn more complex relationships efficiently.

In the LSTM model, the input layer serves as the entry point, where time series data is initially fed into the model. This layer is responsible for transforming the raw data into a format that can be interpreted by the model for training purposes. At this stage, each time step in the time series data is mapped to a vector and subsequently input into the model. In the context of the plastic packaging industry, data such as sales quantities, inventory levels, and seasonal factors are passed into the input layer. The input layer processes this information and provides the foundational data required for the hidden layers to learn long-term dependencies (Keskin et al., 2022). Furthermore, normalization procedures are applied within the input layer to preserve the sequential structure of the time series data and ensure the model functions correctly. Data normalization scales the input

data to a standardized range, allowing the model to process data of varying magnitudes effectively. In the plastic packaging industry, these preprocessing steps enhance the accuracy of sales data handling, thereby improving the overall performance of the model.

The learning process in LSTM primarily occurs within the hidden layers (Khokhar et al., 2023). Hidden layers are crucial to the model's function, as they learn the dependencies embedded in the input data. These layers house the memory cells and gate mechanisms of the LSTM, enabling the model to extract meaningful information from past data. Memory cells retain the information over long time periods, allowing the model to capture long-term dependencies within the time series. A critical hyperparameter influencing the model's complexity and learning capacity is the number of hidden layers. In the plastic packaging industry, where processes such as demand forecasting and production planning are inherently complex, the use of multiple hidden layers facilitates deeper learning. Each hidden layer processes the information it receives from the previous layer and passes it on to the next, thereby refining the model's ability to learn intricate patterns in the time series data (Keskin and Koçoğlu, 2022; Keskin et al., 2022). The activation functions employed in the hidden layers enable the model to learn non-linear relationships. The sigmoid activation function determines the relevance of the input data by scaling the values between 0 and 1, while the tanh activation function scales the cell state between -1 and 1, allowing for more precise processing of information. In the plastic packaging industry, hidden layers, through these activation functions, learn the complex relationships present in production data and uncover long-term trends (Leong, 2023).

The output layer of the LSTM model, the last step, will provide an avenue for the model to state the forecasted results. This layer will process the information from the hidden layer and generate an output for the target variable. In general, most time series analysis involves outputs that represent forecasts for one or more future time steps. In the output layer, the activation functions are utilized for scaling purposes and in giving meaning to the model forecasts. In time series forecasts, linear activation functions are generally preferred, and the model's output values are ensured to be on the same scale as the actual values. The output layer used in sales forecasting of the plastic packaging industry makes these forecasts reliable and applicable. While the input layer converts the time series data into a processable form, the hidden layers learn the dependencies from this data and the output layer produces meaningful forecasts (Malashin et al., 2024). In areas with dynamic and complex data structures such as the plastic packaging industry, the interaction between these three layers enables the model to make high-accuracy predictions and improve the strategic decision-making processes of businesses. The structure of the LSTM neural network is shown in Figure 1.

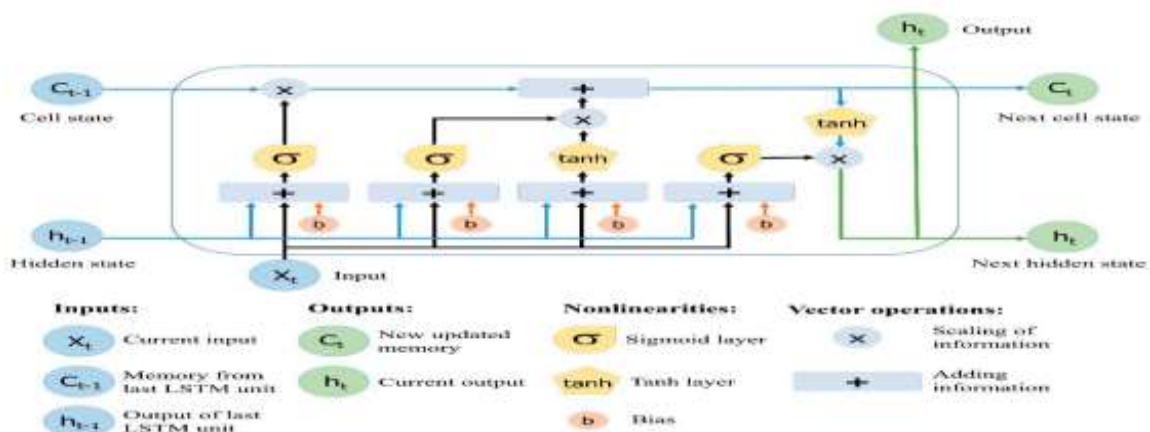


Figure 1. The structure of the Long Short-Term Memory (LSTM) neural network (Yan,2018)

The first step in constructing an LSTM network involves determining which information should be discarded from the cell state at a given time step. This selection process is governed by the sigmoid activation function, which receives as inputs the previous hidden state h_{t-1} , and the current input x_t . The sigmoid function

subsequently determines the proportion of the previous cell state C_{t-1} that should be forgotten, producing the forget gate vector f_t , whose values lie between 0 and 1 (Le et al., 2019). This mechanism is expressed mathematically in Equation (1):

$$f_t = \sigma(w_f[h_{t-1}, X_t] + b_f) . \quad (1)$$

Herein, σ represents the sigmoid function, and W_f and b_f are the weight matrices and bias, respectively, of the forget gate.

Following this, the model determines which new information from the current input X_t should be incorporated into the cell state. This process comprises two components: a sigmoid layer that identifies which values should be updated and a tanh layer that creates candidate values scaled between -1 and 1 . The outputs of these two components are subsequently combined to update the cell state, as illustrated in Equations (2–4) (Le et al., 2019):

$$i_t = \sigma(w_i[h_{t-1}, X_t] + b_i), \quad (2)$$

$$N_t = \tanh(w_n[h_{t-1}, X_t] + b_n), \quad (3)$$

$$C_t = C_{t-1} f_t + N_t i_t. \quad (4)$$

Here, C_{t-1} and C_t represent the cell states at times $t - 1$ and t , respectively, while W and b are the weight matrices and bias of the cell state.

In the final stage, the LSTM calculates the output vector h_t , which is derived from the updated cell state but filtered through an output gate. First, the sigmoid function determines which parts of the cell state should contribute to the output. Then, this output gate vector O_t is multiplied by the tanh-transformed cell state to generate the final hidden state, as expressed in Equations (5–6) (Le et al., 2019):

$$O_t = \sigma(w_o[h_{t-1}, X_t] + b_o), \quad (5)$$

$$h_t = O_t \tanh(C_t). \quad (6)$$

Here, W_o and b_o are the weight matrices and bias, respectively, of the output gate.

Model performance is evaluated in detail on the test data using four key metrics: RMSE, MAE, MSE, and MAPE. These metrics are essential for assessing both the magnitude and the overall nature of the model's estimation errors. While RMSE and MAE provide insights into the accuracy of the model's predictions and the magnitude of error rates, MAPE expresses the magnitude of the errors as a percentage, and MSE reflects the mean squared error. These metrics are crucial for evaluating the model's performance and identifying discrepancies in its predictions (Rahimov, 2024).

MAE expresses the average of the absolute differences between the values predicted by the model and the actual values. This metric is used to measure the magnitude of the prediction errors, and low MAE values indicate that the model's predictions are close to the actual values. MAE is a simple and understandable method for evaluating prediction accuracy (Rahimov, 2024). The calculation of MAE is shown in Equation 7.

$$\text{MAE} (y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (7)$$

In Equation 7, y_i represents the actual values, $\hat{y}_i$ represents the estimated values, and n represents the total number of data.

MSE represents the average of the squared differences between the model's predicted values and the actual values. This metric is used to measure the magnitude of the prediction errors and to evaluate the model's performance. Low MSE values indicate that the model's predictions are closer to the actual values, which indicates that the model can make high-accuracy predictions (Rahimov, 2024). The calculation of MSE is shown in Equation 8.

$$\text{MSE} (y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (8)$$

In Equation 8, y_i represents the actual values, $\hat{y}_i$ represents the estimated values, and n represents the total number of data.

RMSE is the square root of the mean square of the difference between the values predicted by the model and the actual values. This metric evaluates the model's performance by measuring the magnitude of the prediction errors. Lower RMSE values indicate that the model's predictions are closer to the actual values, indicating that the model is able to make high-accuracy predictions (Rahimov, 2024). The calculation of RMSE is shown in Equation 9.

$$\text{RMSE}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n-1} (y_i - \hat{y}_i)^2} \quad (9)$$

In Equation 9, y_i represents the actual values, $\hat{y}_i$ represents the estimated values, and n represents the total number of data.

MAPE is the average of the absolute differences between the values predicted by the model and the actual values expressed as a percentage compared to the actual values. This metric measures the magnitude of the prediction errors as a percentage, and low MAPE values indicate that the model's predictions are close to the actual values. MAPE is widely used to evaluate the accuracy of the model (Rahimov, 2024). The calculation of MAPE is shown in Equation 10.

$$\text{MAPE}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^{n-1} \frac{|y_i - \hat{y}_i|}{|y_i|} \times 100 \quad (10)$$

In Equation 10, y_i represents the actual values, $\hat{y}_i$ represents the estimated values, and n represents the total number of data.

The models with MAPE values below 10% are classified as having “high accuracy” and the models between 10% and 20% are classified as “accurate predictions” (Witt and Witt, 1992)

According to classification thresholds proposed by Witt and Witt (1992) and Lewis (1982), models with MAPE values below 10% are considered to provide “very good” or “high-accuracy” forecasts. Values ranging from 10% to 20% are typically interpreted as “good” or “accurate,” providing a practical benchmark for evaluating forecasting models.

Although alternative evaluation metrics such as the Symmetric Mean Absolute Percentage Error (SMAPE) and the Coefficient of Determination (R^2) are prevalent in the literature, they were intentionally excluded from this study due to limitations in the context of time series modeling with LSTM architectures.

SMAPE is often cited as an alternative to MAPE, particularly in scenarios where actual values approach zero. However, it tends to produce disproportionately high error values under such conditions, which may lead to misleading interpretations. This issue is especially relevant in periods of low demand within the plastic packaging industry. Furthermore, despite its mathematical appeal, SMAPE lacks the intuitive interpretability of MAPE, particularly in business environments where percentage-based error metrics are commonly used.

R^2 , while a standard metric in traditional regression analysis, is not well-suited for time series forecasting using neural networks. It measures the proportion of variance explained by the model but does not directly reflect the magnitude of forecast errors. Additionally, due to the presence of autocorrelation, seasonality, and trends in time series data, R^2 may give an overly optimistic impression of model performance without accurately assessing predictive accuracy.

Given these limitations, the metrics selected for this study—MAE, MSE, RMSE, and MAPE—offer a more appropriate and comprehensive framework for evaluating model performance. These metrics enable a clear and interpretable assessment of forecast accuracy, error distribution, and practical relevance in the context of demand forecasting for the plastic packaging sector.

The effectiveness of the LSTM model in time series forecasting is highly dependent on robust data preprocessing and normalization techniques. Preprocessing involves handling issues such as missing values, outliers, and

varying data scales to ensure the dataset is clean and structured appropriately for modeling. Normalization is particularly critical for LSTM networks, as they are sensitive to differences in variable scales. By rescaling the data—typically using Min-Max normalization to a range between 0 and 1—the model can process inputs more effectively and converge faster during training.

This step is especially important in sectors like plastic packaging, where variables such as production costs and sales volumes may exist on very different scales. Proper normalization ensures balanced learning and contributes to improved prediction accuracy and generalization.

For that reason, the dataset selected is in a time series with records taken from the plastic packaging industry, which will give information regarding levels of future demands. To enable fitting these into the model appropriately, there has been a preliminary data normalizing process first, allowing the model to study it efficiently and effectively; then the time series has been prepared using the Min-Max scaling method with an encoding range of 0-1 (Rahimov, 2024). The scaling process is performed using Equation 11:

$$X_t = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (11)$$

Before modeling the data, it needs to be reshaped using a certain time step. The time step allows the model to predict future sales based on past data. The data is divided into series using a specific time step. For each time series, input (X) and output (Y) sets are defined as found in Equation 9. The input set contains the historical data that the model needs to make a prediction, while the output set contains the sales information that needs to be predicted. This method enables the model to learn sequential relationships between the data. It also helps to capture factors such as trend and seasonality in time series data. Thus, the model can make better and more reliable predictions (Rahimov, 2024). The creation of input and output sets is shown in Equation 12:

$$X_t = [X_1, X_2, \dots, X_{t-1}] Y_t = X_t \quad (12)$$

In Equation 12, X_t represents the input data at time step t , while Y_t denotes the sales value that the model aims to predict. Each set X_t includes data points from a specific time period (n days), and Y_t corresponds to the closing price on the day following this time period. This approach enables the model to learn to predict the price for the $n+1$ -th day based on the data from the previous nnn days. The normalized and reformatted data are then split into training and test sets, with typically 80% allocated for training and 20% for testing. This division is essential to evaluate the model's performance on unseen data. The training set is utilized for model learning, whereas the test set is employed to assess the model's predictive accuracy and its generalization capabilities. Data preprocessing steps serve as the foundation for both training the model and evaluating its performance (Rahimov, 2024).

Separating seasonality and trend components from the data is another crucial preprocessing step. Since long-term trends and seasonal patterns can complicate the learning process, these components need to be extracted before inputting the data into the model. In the plastic packaging industry, for instance, demand patterns that fluctuate seasonally can be identified and removed, allowing LSTM to focus on learning the underlying trends (Ucal, 2022; Wang et al., 2022).

Moreover, in time series forecasting, the sequential nature of the data must be preserved. Properly aligning time steps is essential for LSTM to learn from the data in a chronological order, which is particularly important when forecasting demand in the plastic packaging sector. For example, aligning weekly or monthly demand data correctly allows the model to more easily capture long-term dependencies (Yamak et al., 2019).

Early stopping and regularization techniques are also employed during training to prevent overfitting. Early stopping halts training when the validation loss starts increasing, ensuring the model generalizes well to new data. This prevents the model from becoming overly complex and ensures it remains effective for future predictions (Ucal, 2022).

Maintaining the sequential structure of the time series is crucial. LSTM models learn long-term dependencies by processing data over time steps, using a sliding window of past data. The window size defines how much historical data the model considers, and adjusting this window can enhance prediction accuracy. For the plastic packaging industry, window sizes based on annual or monthly periods are common, particularly for seasonal demand forecasting.

The loss function, typically the MSE, plays a critical role in guiding the learning process by quantifying the difference between the model's predictions and actual values. LSTM models aim to minimize this loss, improving prediction accuracy.

Two key hyperparameters in training an LSTM model, epoch and batch size, significantly influence its performance in forecasting tasks. Both parameters affect the model's learning process and its ability to generalize from training data to make accurate predictions.

An epoch represents the number of times the model processes the entire training dataset. Each epoch involves updating the model's weights to minimize the loss function. A higher number of epochs allows the model to learn patterns in more detail, but if set too high, it can lead to overfitting, causing the model to become overly specialized in the training data and lose its ability to generalize to new data. (Abbasimehr and Paki, 2021; Akçay and Yiltas-Kaplan, 2024). In the plastic packaging industry, careful tuning of epochs is essential for capturing seasonal patterns in demand data. For example, a sufficient number of epochs is needed to learn historical sales trends, but too many may degrade the model's performance on unseen data. Techniques like early stopping are often used to automatically determine the optimal number of epochs by halting training when the validation loss begins to rise (Albeladi et al., 2023).

Batch size refers to how the training data are divided into smaller subgroups, with each batch processed individually during training. Smaller batch sizes lead to more frequent weight updates, allowing for finer learning, while larger batch sizes speed up training but can result in less precise updates (Anwar et al., 2024). In the plastic packaging industry, the choice of batch size depends on the structure of the data and the goal of the model. For instance, small batch sizes are beneficial for learning finer patterns, especially when dealing with smaller time series data, such as daily sales. On the other hand, larger batch sizes can be useful for models working with large datasets, like monthly or annual sales data, as they improve computational efficiency. However, small batch sizes generally offer better generalization ability, as they expose the model to more variable data in each update, leading to more accurate predictions on test data (Aytaç et al., 2022).

4. Constructing the LSTM Model

In this study conducted for a plastic company located in Gaziantep, an LSTM model was developed in Python using sales data collected every 15 days from January 1, 2015, to December 16, 2024. A total of 240 data points were utilized in the analysis. Furthermore, the data used in this study can be obtained from the corresponding author upon request, as its inclusion would disrupt the page's integrity.

The 'sales' data is normalized to a range between 0 and 1, which is important for neural networks because activation functions like tanh and sigmoid perform more effectively with normalized data. Once the data is normalized, it creates input-output pairs for the model. Given a sequence of data points (time steps), it uses the last sequence_length data points as input (X) and the next data point as the target (Y). For example, if sequence_length is set to 12, the model learns from the last 12 data points and predicts next sales. Data preprocessing has been completed, as illustrated in Figure 2.

```
# Data normalization using MinMaxScaler
scaler = MinMaxScaler(feature_range=(0, 1))
data_scaled = scaler.fit_transform(data[['Sales']])

# Sliding window function to create sequences for LSTM
def create_dataset(data, sequence_length=12):
    X, y = [], []
    for i in range(sequence_length, len(data)):
        X.append(data[i-sequence_length:i, 0]) # Last 'sequence_length' data points for X
        y.append(data[i, 0]) # The next data point as the target (y)
    return np.array(X), np.array(y)

# Prepare the dataset with the specified sequence length
sequence_length = 12 # Period length (e.g., 12 months)
X, y = create_dataset(data_scaled, sequence_length=sequence_length)
```

Figure 2. Data Preprocessing for LSTM models

In this study, a 70%–30% split between the training and test datasets was adopted, resulting in 168 data points for training and 72 for testing. While an 80%–20% split is commonly used in many machine learning applications, the choice of a 70%–30% distribution was made to ensure a more comprehensive evaluation of the model's forecasting performance. Since the dataset consists of 240 time-dependent data points spanning nearly a decade, allocating a larger portion (30%) to the test set allows the model's generalization capability to be assessed more robustly across recent and potentially more volatile periods. In time series forecasting, especially in industries affected by seasonal and trend-based fluctuations like plastic packaging, having a larger test set helps reveal how well the model adapts to unseen future patterns. This trade-off between training and evaluation allows for more reliable model validation without significantly compromising the training depth.

Figure 3 presents the partitioning of the dataset into training and testing subsets, which constitutes a fundamental step in the development and validation of the forecasting model. As depicted in the figure, 70% of the observations were assigned to the training set, while the remaining 30% were allocated to the testing set, thereby ensuring an unbiased evaluation of the model's predictive performance on unseen data.

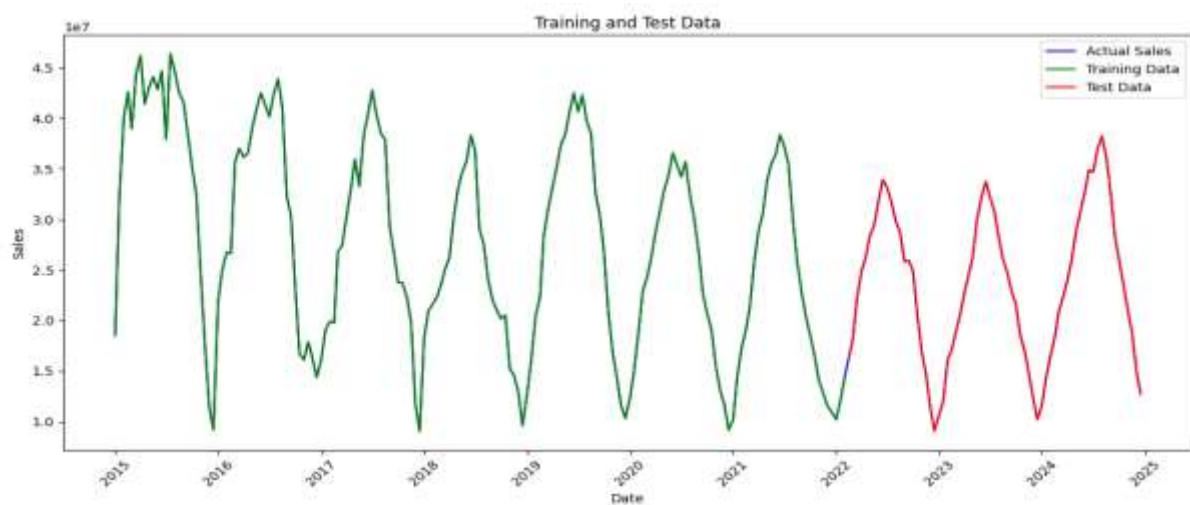


Figure 3. Separation of data for training and testing

The architecture of the LSTM model comprises two sequential LSTM layers, each consisting of 100 units. The first LSTM layer is configured with the parameter `return_sequences=True`, enabling the generation of an output vector for each time step. This configuration is essential for the hierarchical stacking of LSTM layers, as it allows the subsequent layer to process a sequence rather than a single aggregated output. The second LSTM layer is set with `return_sequences=False`, indicating that only the final output corresponding to the last time step is returned. This design choice reflects the cessation of further LSTM stacking and prepares the output for transition to subsequent dense or regularization layers.

To address the risk of overfitting, Dropout regularization is incorporated into the model architecture. The first Dropout layer enhances the model's generalization capability by randomly deactivating 20% of input units during the training phase, thereby preventing the network from over-reliance on specific neurons. An additional Dropout layer with the same 20% dropout rate is integrated after the second LSTM layer to further strengthen regularization and promote robust learning across varying data patterns. Collectively, these regularization strategies contribute to improving the stability and predictive reliability of the LSTM model.

The Dense layer consists of 50 units and utilizes the ReLU (Rectified Linear Unit) activation function. The ReLU function facilitates the model's ability to learn non-linear relationships by setting negative values to zero and only allowing positive values to pass through.

The final layer is a Dense layer with a single unit, which outputs a continuous value for the regression task. This layer is responsible for generating the model's predictions. The LSTM model is illustrated in Figure 4.

```
# Build the LSTM model
model = Sequential([
    LSTM(100, return_sequences=True, input_shape=(X_train.shape[1], 1)), # First LSTM layer with 100 units
    Dropout(0.2), # Dropout layer for regularization
    LSTM(100, return_sequences=False), # Second LSTM layer with 100 units
    Dropout(0.2), # Dropout layer for regularization
    Dense(50, activation='relu'), # Dense layer with 50 units and ReLU activation
    Dense(1) # Output layer with 1 unit for regression output
])
```

Figure 4. Build the LSTM model

The Adam optimization algorithm was chosen for training the model. Adam is an adaptive optimization method that automatically adjusts the learning rate during the training process, resulting in faster convergence and a more efficient learning process. In this study, the learning rate is set to 0.001.

Batch size refers to the number of samples the model processes at each iteration during training. In this case, the batch size is set to 8, meaning that the model will update its weights after processing 8 samples at a time.

The number of epochs refers to the number of complete passes through the entire training dataset. One epoch corresponds to one full cycle through all the training data. For this model, training is conducted over 50 epochs, meaning the model will iterate over the entire training dataset 50 times to learn and improve its performance.

5. Results and Discussion

The success of the developed LSTM model for monthly sales forecasting in a plastic company's mold production lies in its ability to capture long-term dependencies. LSTM layers are highly effective in learning long-term relationships and can model complex patterns in historical data. In the studies reviewed, data was typically split with 80% for training and 20% for testing. However, in this study, which utilized 240 data points, the data was divided as 70% for training and 30% for testing.

The model's success is evaluated through various error metrics, including MAE, MSE, RMSE, and MAPE. MAE directly measures the magnitude of the errors and gives equal weight to all errors. A low MAE value indicates that the model's predictions are close to the actual values. MSE amplifies the error magnitudes by squaring them, which results in greater penalties for large errors. A lower MSE suggests that the model's predictions are more accurate and closer to the real data.

RMSE, a derivative of MSE, provides an error metric with the same units as the predicted values, making it more interpretable. A low RMSE indicates that the model's errors are minimal. MAPE averages the percentage errors between the predicted and actual values, providing a useful metric for understanding the overall accuracy of the model. MAE and MAPE are generally more intuitive, as they directly reflect the size of the errors, either absolutely or in percentage terms.

While MSE and RMSE penalize larger errors more heavily, making them suitable when the focus is on minimizing large errors, MAE and MAPE offer a more straightforward understanding of the model's accuracy. A low RMSE and MAE suggest that the model's predictions are very close to the actual values, and small prediction errors have occurred. Additionally, a low MAPE value further demonstrates the model's success, as it indicates that the prediction errors are low in percentage terms.

Train MAE: 1402000.42029

Train MSE: 2720005105705.60889

Train RMSE: 1649243.79814

Train MAPE: 0.07160

Test MAE: 1675830.05797

Test MSE: 3859636355239.13037

Test RMSE: 1964595.72310

Test MAPE: 0.08797

A MAPE (Mean Absolute Percentage Error) value of 7.16% suggests that the model's predictions on the training dataset are generally accurate; however, there is still an error margin of 7.16%. This means that, on average, the model's predictions deviate from the actual values by 7.16%.

The MAPE on the test data represents the percentage error of the model's predictions relative to the true values in the test set. Specifically, the model's predictions on the test data deviate from the true values by 8.79%.

The model performs well on the training data, as indicated by the relatively low MAE and MAPE values, which suggest that the model is able to make accurate predictions during training.

However, the errors on the test data are larger. The MSE, RMSE, and MAPE values for the test set are higher compared to those of the training set, which indicates that the model performs slightly worse on the test data and faces some challenges in generalizing to unseen data.

In conclusion, while the model's predictions are generally accurate, larger errors are observed on the test data. The MAPE value of 7.16% indicates that the model is performing with reasonable accuracy, although there is room for further improvement in its predictive performance.

In the finalized model, monthly sales forecasts for the year 2025 were generated. The sales estimates for 2025 are presented in Table 1.

Table 1: Monthly sales forecasting for 2025

Month	Sales
January	13418849
February	13462072
March	14214780
April	15775449
May	17675234
June	19987530
July	22285204
August	24492594
September	26486854
October	28175294
November	29488898
December	30377402

Figure 5 illustrates four key lines derived from different data sets to explore the relationship between predicted sales data and actual sales figures. The blue line represents the actual historical sales data used by the model to make its forecasts. This line highlights the temporal changes in sales and seasonal trends, providing insight into the fluctuations observed over time. Real sales data often exhibit seasonal patterns, reflecting periods of increase or decrease, as well as annual, monthly, or daily variations. The blue line serves as a baseline for evaluating the accuracy of the model's predictions, allowing for a direct comparison with past data.

The green line shows the predictions made by the model on the training data. Training data consists of historical data used during the model's learning phase, and is crucial for assessing how well the model performs in predicting known outcomes. While the green line is close to the actual sales data, minor differences are observable, indicating that the model is performing effectively on the training data and capturing the underlying patterns in past sales. The proximity between the model's predictions and the actual training data demonstrates the model's ability to learn and make forecasts based on historical trends.

Test data, which is typically separated from the training data, serves to evaluate the model's ability to generalize to new, unseen data. The red dashed line represents the model's predictions on the test data, illustrating how well the model can apply its learned patterns to new data. This line is critical for assessing the model's predictive accuracy and generalization capacity. The blue dashed line shows the actual test sales, providing a point of comparison with the model's test predictions. The differences between the red and blue dashed lines offer valuable information regarding the model's performance on the test set and provide insights into its ability to generalize beyond the training data.

The purple line represents the model's predictions for 2025. These predictions are projections based on past sales data, and they demonstrate the model's ability to forecast future sales. The 2025 forecasts are particularly valuable for strategic decision-making within the company. While these future predictions generally follow the same trends as historical data, deviations are observed in certain periods. These deviations highlight areas for potential improvement, offering an opportunity to refine the model's accuracy and enhance its capacity to predict future trends more precisely.

In conclusion, the graph illustrates that the model successfully predicts sales trends consistent with historical data, offering useful insights into future sales projections. However, deviations in the 2025 predictions point to the need for further model refinement and optimization. Such adjustments would improve the model's accuracy and enhance its generalization power. These findings underscore the model's role in providing reliable forecasts, which are essential for making informed strategic decisions in the business context.

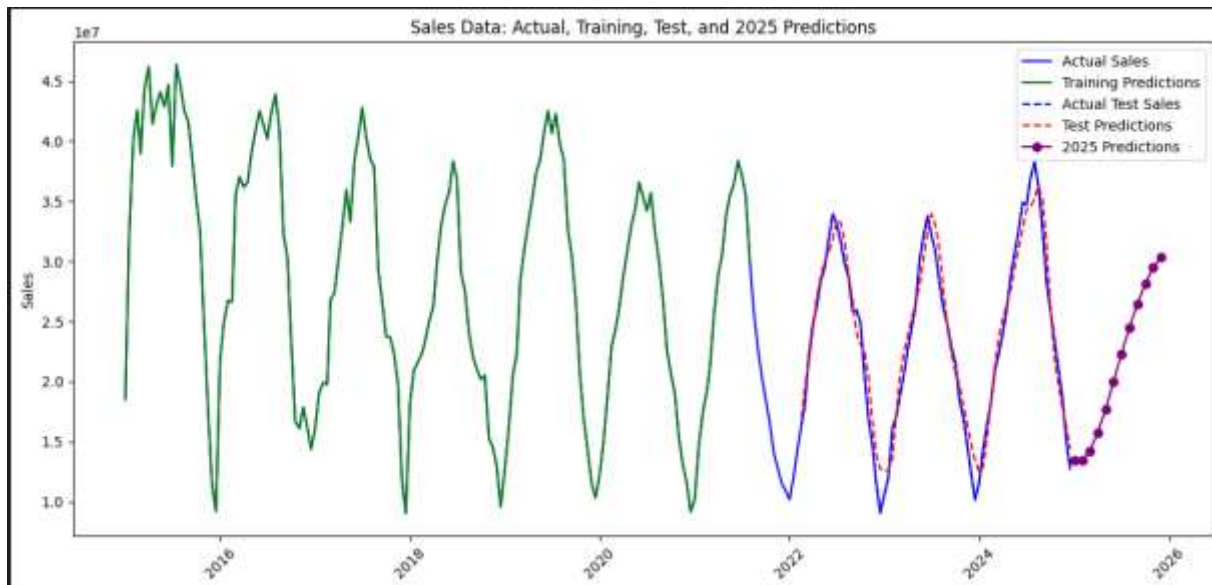


Figure 5: Comparing the model's past performance with future forecasting

Figure 6 presents a comparison between the forecasted data and the actual sales data. The blue line represents the historical sales data, extending backwards to provide a clearer view of how these data have evolved over time. Actual sales data often exhibit seasonal patterns and trends, along with fluctuations on an annual, monthly, or daily basis. These fluctuations reflect periods when sales either increase or decrease.

The red line illustrates the model's predictions based on the training and test data, including the 12-month sales forecast for 2025. These forecasts reflect the model's ability to predict future sales based on past data. While the

predictions generally follow the overall trends, some deviations are evident when comparing them with the actual sales figures.

Upon examining the visualized data, it is observed that the differences between the forecasts and the actual data are generally small. However, there are instances where deviations occur in the model's forecasts during certain periods. This serves as an important indicator for assessing the model's accuracy and its ability to generalize to new data.

The model's 2025 forecasts, derived from historical sales data, play a crucial role in gaining a deeper understanding of potential sales trends. Accurate predictions for 2025 provide a valuable reference for making informed decisions within the company.

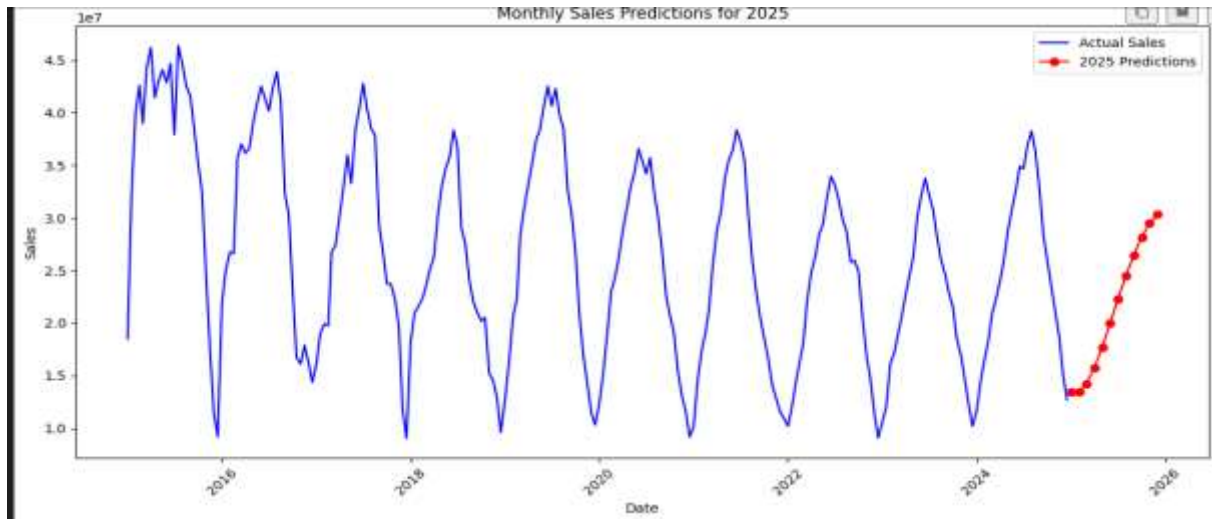


Figure 6: Comparing the model's actual sales with future forecasting

Figure 7, which visualizes the monthly sales projections for 2025, offers valuable insights into the anticipated sales trends over the course of the year. The sales estimates for 2025 are represented by a red line, with each point marked by circular markers corresponding to the estimated sales figures for each month. This line illustrates the temporal progression of sales and the overall trend expected to unfold throughout the year.

The sales projections for 2025 reveal a significant upward trend in the latter months of the year. For instance, sales are forecasted to start at approximately 13.4 million in January and rise to 30.4 million by December. This marked increase indicates a strong expectation of growth in sales over the remainder of the year.

Another notable feature in Figure 7 is the consistent upward trajectory of sales throughout the year. Although sales begin at lower levels, particularly in the first half of the year (January-May), a sharp rise is observed in the second half (June-December). This surge could be attributed to factors such as seasonal demand fluctuations, strategic adjustments, or shifts in market dynamics. The more pronounced growth observed toward the final quarter of the year may be explained by seasonal effects, the holiday season, or the influence of year-end targets. However, the exact causes of this increase would benefit from further analysis and a more comprehensive dataset.

The continuous upward trend depicted in Figure 7 underscores the strong growth expectation embedded in the 2025 sales projections. This growth trend likely reflects both the company's expansion goals and the anticipated increase in market demand. However, to better assess the accuracy and reliability of these projections, comparisons with actual sales data will be crucial.

Figure 7 serves as an essential tool for strategic planning and decision support systems. By integrating these projections, the company can make informed decisions to achieve its sales targets for 2025 and determine appropriate strategies for navigating the market. Moreover, the graph provides an important reference for visualizing future sales expectations and supports the ongoing process of reviewing and adjusting the company's annual objectives.



Figure 7: Monthly Sales Predictions for 2025

6. Conclusion

This study aimed to evaluate the effectiveness of the LSTM model in the context of sales forecasting using real-world data from a plastic manufacturing company based in Gaziantep. The experimental results demonstrate that the LSTM model significantly outperformed traditional statistical models such as ARIMA and SARIMA, as well as the GRU deep learning model. With MAPE values of 7.16% for the training set and 8.79% for the test set, the LSTM model achieved a level of accuracy that is considered highly reliable for practical applications in sales forecasting.

In contrast, the ARIMA, SARIMA, and GRU models exhibited notably higher test error rates (MAPE above 20%), with ARIMA and SARIMA showing considerable performance drops between training and test sets—indicative of overfitting. While the GRU model performed slightly better, it still failed to produce test results below the 10% MAPE threshold. These outcomes ultimately justified the selection of the LSTM model as the core predictive framework for this study.

Several enhancements were applied to the LSTM model to ensure its generalization capability and prevent overfitting, including the use of dropout layers and carefully tuned hyperparameters. Dropout layers helped regularize the training process by randomly deactivating neurons, thus forcing the model to learn generalized patterns rather than memorizing training data. The minimal discrepancy between training and test error metrics further supports the conclusion that the LSTM model successfully generalized to unseen data.

Moreover, the findings have practical implications for production and inventory planning. Accurate sales forecasts can assist in maintaining optimal stock levels, minimizing surplus inventory, and improving overall operational efficiency. This, in turn, is expected to reduce costs and enhance customer satisfaction.

However, several limitations of the study should be acknowledged:

6.1 Limitations

Lack of Seasonal Components:

Although the actual sales data revealed seasonal patterns—such as higher sales during summer months and lower sales during winter—these trends were not captured by the LSTM model. Instead, the model produced a linear growth pattern throughout the year. This highlights the need to incorporate seasonality into future forecasting models.

Limited Product Granularity:

The current model was developed for general sales forecasting of a single preform mold type, without differentiating between product categories (e.g., colored vs. transparent, customer-specific products, or raw material variations). More detailed forecasting by product type would enhance the model's utility in production planning.

Exclusion of External Factors:

Influential external variables, such as holidays, promotions, market fluctuations, or economic indicators were not included in the dataset. Integrating such exogenous variables could significantly improve the model's responsiveness and accuracy.

Limited Dataset Scope and Generalizability:

The model was trained and tested on data from a single company, which may limit its generalizability to other industries or geographic regions. Additionally, the analysis was based on **monthly sales data**, using **weekly or daily data** could offer better resolution for short-term forecasting and reveal subtler patterns.

Evaluation Based Solely on Numerical Metrics:

While the study employed quantitative error metrics (MAPE, MAE, RMSE, etc.) to assess model performance, it did not evaluate qualitative aspects such as user satisfaction, decision-making impact, or business outcomes. These dimensions could provide further insight into the real-world applicability of the model.

These limitations can be addressed in future research to further enhance the model's prediction accuracy and generalization capability. In particular, integrating seasonal ARIMA (SARIMA) components into hybrid LSTM models, incorporating exogenous variables, and developing product-level forecasting approaches could significantly improve sales forecasting performance.

In conclusion, this study demonstrates that the LSTM model holds strong potential in sales forecasting applications and can be effectively utilized as a decision-support tool across various operational areas—from production planning to inventory optimization. The model's adaptable structure provides a solid foundation for future enhancements and research.

References

- Abbasimehr, H., and Paki, R. (2021). Improving time series forecasting using LSTM and attention models. *Journal of Ambient Intelligence and Humanized Computing*, 13(1), 673-691.
- Akçay, H., and Yiltas-Kaplan, D. (2024). Zaman serileri tahminleme algoritmaları ile kontör tüketim tahminlemesi ve karşılaştırmalı uygulaması. *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, 27(1), 166–189.
- Albeladi, K., Zafar, B., and Mueen, A. (2023). Time series forecasting using LSTM and ARIMA. *International Journal of Advanced Computer Science and Applications*, 14(1), 313-320.
- Anwar, M. A., Suprihatin, S., Sasongko, N. A., Najib, M., and Pranato, B. (2024). Challenges and prospects of multilayer plastic waste management in several countries: A systematic literature review. *Case Studies in Chemical and Environmental Engineering*, 10, 100911.
- Aytaç, U. C., Küçükyılmaz, T., and Tarakçıoğlu, G. S. (2022). Comparison of time series models for predicting online gaming company revenue. *Journal of Statistics & Applied Science*, (6), 26–36.
- Bachu, Y. (2019). *Packaging demand forecasting in logistics using deep neural networks* [Master's thesis, Blekinge Institute of Technology]. DiVA Portal.

- Çalık, S. (2025). *Core network anomaly detection using the Lstm model and comparison with various unsupervised learning methods* [Master's thesis, İstanbul Teknik Üniversitesi]. Büyük Veri ve İş Analitiği Ana Bilim Dalı.
- Dinata, M. C., and Suharjito, S. (2024). Materials inventory optimization using various forecasting techniques and purchasing quantity in packaging industry. *Journal of Industrial Engineering and Management*, 17(2), 321–343.
- Erdal, G. (2020). İlaç sanayinde kullanılan plastik ambalajlar. *Uluslararası Sanat Tasarım ve Eğitim Dergisi (USTAD)*, 5(1), 25-33.
- Eraslan, İ. H., Karataş, A., and Kaya, H. (2007). Türk plastik sektörünün rekabetçilik analizi. *İstanbul Ticaret Üniversitesi Sosyal Bilimler Dergisi*, 6(11), 203-219.
- Fatima, S. S. W., and Rahimi, A. (2024). A review of time-series forecasting algorithms for industrial manufacturing systems. *Machines*, 12, 380.
- Gür, Y. E., & Eşidir, K. A. (2023). Yapay sinir ağları ile Türkiye plastik sektörü ithalat tahmini: 2023 yılı nisan-aralık ayları. *Akademik Hassasiyetler*, 10(23), 91–114.
- Işık, D. N. (2023). *Zaman serileri tahmininde melez bir yaklaşım* [Yüksek lisans tezi, Fatih Sultan Mehmet Vakıf Üniversitesi]. Bilgisayar Mühendisliği Ana Bilim Dalı.
- Kalinggo, B. A., and Zulkarnain. (2020). Time series forecasting for non-stationary data: A case study of petrochemical product price. In *Proceedings of the 3rd Asia Pacific Conference on Research in Industrial and Systems Engineering, APCORISE '20* (86–92).
- Keskin, B., and Koçoğlu, Ş. (2022). Ambalaj sektörünün sürdürülebilirliği ve petrol bazlı plastik: Plastik ambalaj sektörünün petrole olan bağımlılığının analizi. *Avrupa Bilim ve Teknoloji Dergisi*, 35, 252-258.
- Keskin, F. D., Çiçekli, U. G., and İçli, İ. D. (2022). Prediction of failure categories in plastic extrusion process with deep learning. *Journal of Intelligent Systems: Theory and Applications*, 5(1), 27–34.
- Khokhar, A., Talpur, S., and Memon, M. A. (2023). Comparative analysis of LSTM, BILSTM, and ARIMA for time series forecasting on 116 years of temperature and rainfall data from Pakistan. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, 10(2), 350–357.
- Leong, L. A. (2023). *Forecasting SOXX with long short-term memory (LSTM) neural networks based on varying sampling periods* [Doctoral dissertation, Selinus University]. The Department of Finance & Economics.
- Lewis, C. D. (1982). *Industrial and business forecasting methods*. Butterworths Publishing. London, 40.
- Malashin, I., Tynchenko, V., Gantimurov, A., Nelyub, V., and Borodulin, A. (2024). Applications of long short-term memory (LSTM) networks in polymeric sciences: A review. *Polymers (Basel)*, 16(18), 2607.
- Rahimov, S. (2024). *Derin öğrenme ile borsa fiyatlarının tahmini: LSTM yaklaşımı* [Master's thesis, Ankara Üniversitesi]. Ankara Üniversitesi Enstitüsü.
- Uçal, B. (2022). Plastik Ambalaj Sektör Raporu: Ocak – Aralık 2022. *İstanbul Kimyevi Maddeler ve Mamulleri İhracatçıları Birliği (İKMİB)*.
- Yamak, P. T., Yujian, L., and Gadosey, P. K. (2019). A comparison between ARIMA, LSTM, and GRU for time series forecasting. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence (ACAI'19)* (49–55).
- Yan, S. (2016). Understanding LSTM and its diagrams. *Medium*. <https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714>.
- Le, X.-H., Ho, H. V., Lee, G., and Jung, S. (2019). Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. *Water Resources Management*, 33(12), 4151-4163.
- Tüler, A. (2025). *Implementation Of Long Short-Term Memory (LSTM) Deep Learning Algorithm To Predict Flow Rate And Water Quality Index (WQI)* [Master's thesis, İzmir Ekonomi Üniversitesi]. Bilgisayar Mühendisliği Ana Bilim Dalı.

Wang, C. C., Chang, H. T., and Chien, C. H. (2022). Hybrid LSTM-ARMA demand-forecasting model based on error compensation for integrated circuit tray manufacturing. *Mathematics*, 10(13), 2158.

Witt, S. F., and Witt, C. A. (1992). *Modeling and forecasting demand in tourism*. Academic Press.