10.48047/jocaaa.2025.34.05.41

Intelligent Demand Forecasting in the Bakery Industry Using Convolutional Neural Network Regressors

K Parameswari[1*], S.Yeshwanth[2], K.Yamini[2], G.Sudeep[2], S.Saicharan[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering,
[1,2]Malla Reddy Engineering College and Management Sciences, Kistapur, Medchal, 501401, Telangana, India
*Corresponding author: parameswari@mrem.ac.in

Received: 02.03.2025 Revised: 15.04.2025 Accepted: 25.05.2025

ABSTRACT

Food supply chain management plays a vital role in ensuring the efficient production, distribution, and delivery of food products, particularly in developing economies such as India. However, the sector faces persistent challenges including demand–supply mismatches, inefficient logistics, and excessive food wastage. Reports indicate that nearly 40% of food produced in India is wasted annually, resulting in economic losses of approximately ₹92,000 crores. These issues are further exacerbated by population growth, changing consumer preferences, and market volatility. Traditional demand forecasting methods—such as historical averages, rule-based systems, and manual estimation—fail to capture dynamic factors like seasonality, pricing fluctuations, and promotional effects, leading to overproduction and inefficient resource utilization. To address these limitations, this research proposes a Convolutional Neural Network (CNN) regressor–based demand forecasting framework tailored for the bakery sector. The proposed model effectively extracts spatial and temporal patterns from historical sales and contextual data, enabling accurate weekly demand predictions for bakery products. By capturing complex trends such as seasonal demand shifts and promotional impacts, the system enhances inventory optimization, reduces food wastage, and supports data-driven decision-making. The CNN-based approach highlights the potential of deep learning in improving food supply chain efficiency, sustainability, and economic resilience.

Keywords: Food Supply Chain Management, Demand Forecasting, Convolutional Neural Network, CNN Regressor, Bakery Sector, Inventory Optimization, Food Wastage Reduction, Machine Learning, Logistics Optimization.

1. INTRODUCTION

The global bakery industry is a vital component of the food sector, contributing over USD 500 billion to the global economy, with an estimated compound annual growth rate (CAGR) of 4.6% from 2022 to 2028. This industry encompasses a wide range of products, including bread, cakes, pastries, and cookies, with consumers increasingly demanding fresh, customized, and healthier options. However, despite technological advancements in food processing and packaging, the sector continues to face substantial challenges in managing demand variability and inventory inefficiencies. One of the

10.48047/jocaaa.2025.34.05.41

primary issues remains the perishable nature of baked goods, which leads to either surplus production resulting in wastage or insufficient stock resulting in lost sales.

According to the Food and Agriculture Organization (FAO), approximately 17% of total global food production is wasted, a significant portion of which is attributed to poor demand forecasting and inefficient supply chains in perishable product sectors like bakeries. In retail bakeries, where daily production is based on anticipated consumer demand, inaccurate forecasts often lead to unsold products that must be discarded or sold at a discount. Conversely, underestimation leads to missed opportunities and dissatisfied customers. These inefficiencies affect not only profitability but also contribute to environmental issues such as greenhouse gas emissions from food waste.

Digital transformation and the integration of data analytics have begun to offer solutions to these persistent problems. With vast amounts of historical sales data, customer behavior patterns, weather conditions, and festive seasons affecting consumer choices, the need for precise and adaptive forecasting models is increasingly being recognized. Machine learning and artificial intelligence (AI) models are now playing a vital role in extracting actionable insights, enabling real-time decision-making in production planning and supply chain management. As the bakery sector evolves, data-driven strategies are becoming essential in maintaining competitiveness, sustainability, and customer satisfaction.

2. LITERATURE SURVEY

In their research, [**1**] addressed the challenging task of anticipating demand in assembly industries. The accurate prediction of component demand is crucial, particularly when faced with uncertain end-customer demand. To tackle this issue, they suggested a comprehensive approach that integrates the autoregressive moving average model with exogenous inputs (ARIMAX) and machine learning (ML) models. They also conducted a comparison between these advanced methods and traditional univariate benchmarks.

Ref. [**2**] addressed the issue of demand forecasting in supply chain management in their study, where they presented an advanced model that combines ML techniques, time series analysis, and deep learning methodologies. Using data from the SOK market in Turkey, a rapidly expanding retail chain, they showcased the efficacy of this model in optimizing stocks, cutting costs, and enhancing sales and customer loyalty. The model attains an average of 24.7% of MAPE. The distinctive feature of their study lies in their combination of deep learning methodology and innovative decision integration techniques.

In [**3**] study, the integration of machine learning techniques, notably ARIMAX and neural networks, was explored with the aim of enhancing demand forecasting within multi-stage supply chains. The research showed significant improvements in operational and financial performance, proving the

10.48047/jocaaa.2025.34.05.41

effectiveness of machine learning-based methods in reducing inventory costs and enhancing asset efficiency, especially in capital-intensive industries like steel manufacturing. Additionally, the research emphasized the hybrid forecasting model's ability to capture complex relationships among variables at the aggregate level, resulting in more accurate demand predictions.

Ref. [**4**] introduced the UNISON data-driven framework for forecasting intermittent demand for electronic components in the supply chain. They acknowledged the complexities of this task, which are exacerbated by demand fluctuations, short product life cycles, and long production and technology migration timelines. The framework aims to minimize demand series fluctuations while considering the limitations of available information and the need for cross-product modeling variability. The results demonstrated its effectiveness in improving forecast accuracy to 78.51% for various product categories, thereby enabling flexible decision-making and enhancing supply chain resilience for smart production.

Ref. [**5**] introduced a technique for demand forecasting in highly competitive business environments. Their method, which utilizes a multi-layer long short-term memory (LSTM) network, excels in its ability to dynamically select models. This is achieved through a grid search approach that considers different combinations of LSTM hyperparameters. The method's main strength lies in its capacity to effectively capture nonlinear patterns in time series data, outperforming established forecasting techniques such as ARIMA, ETS, ANN, KNN, RNN, SVM, and single-layer LSTM. Statistical tests confirm the significant superiority of the proposed approach, particularly when assessing performance measures like RMSE and SMAPE, attaining values of 2.596 and 0.1023, respectively. When comparing computational intelligence techniques like SVM, ANN, and LSTM with traditional statistical methods (ETS and ARIMA), it becomes evident that the former outperforms the latter.

Ref. [**6**] introduced the application of distributed XGBoost (DRF) and gradient boosting machine (GBM) learning methods for forecasting backorder situations in the supply chain. They utilized a versatile approach that considers different prediction attributes and the diverse attributes of real-time data influenced by errors from both machines and humans. The approach attains classification accuracy of 79% and 85% for training, which demonstrates its efficiency. This approach gives decision managers the ability to adjust to different scenarios. The authors justified their choice of DRF and GBM models by their ability to provide superior explanations.

Ref. [**7**] examined customer demand prediction for remanufactured products and market development. They highlighted the limitations of traditional statistical models in capturing the nonlinear dynamics of customer demand, particularly in e-marketplaces. To overcome this, they utilized advanced machine learning techniques, such as the XGBoost ensemble regression tree model, to create a precise demand prediction model for remanufactured products. The study used real-world Amazon data and

10.48047/jocaaa.2025.34.05.41

revealed the complexity of predicting remanufactured product demand while demonstrating the effectiveness of machine learning in this field.

Ref. [**8**] developed a machine learning technique to enhance the precision of order demand prediction in e-commerce distribution centers. Their approach integrates the unique traits of time series data and employs an adaptive neuro-fuzzy inference system to analyze the patterns of e-commerce order arrivals as time progresses. The implementation framework consists of four stages to support practical application, and a structured model evaluation framework is in place to ensure the reliability of the predictive model. The model shows RMSE of 8.7, 11.33, and 9.55 for retailers 1, 2, and 3, respectively. Additionally, this method allows for the exploration of integration with optimization or heuristic approaches to enable more robust decision-making in the dynamic e-commerce environment.

Ref. [**9**] developed a manufacturer-and-retailer supply chain model considering the retailer's selling price, product green level, and advertisement effort-sensitive demand. It analyzes a sharing contract policy on manufacturer green investment. This model demonstrates how green production methods and investments can boost manufacturer profits while reducing pollution in the environment. However, the analytical method is more complex and time-consuming.

In a two-layer supply chain management setting with stochastic conditions, Prasanta [**38**] focus on the best replenishment strategies that a retailer implement to maximize average profit when (s)he is presented with trade offers of credit and price discount from suppliers. In order to maximize average profit, the extended study suggests that the merchant borrow half of the cost of purchases from financial support and make payments at the beginning of the advance payment period.

Ref. [**10**] established a production model that screens out faulty products and reduces greenhouse gas emissions. In this case, it has been assumed that the manufacturing process transfers from a controlled state to an uncontrolled state following an arbitrary, random period of time. The research determines the ideal business and production duration, as well as the ideal production rate, and optimizes the model's predicted average profit.

Ref. [**11**] created a flawed production inventory model that includes warranty and insurance coverage for any unintentionally damaged product parts, rework, preventive machine maintenance, and other features. This study determines the best manufacturing schedule and selling price for the product's various warranty periods in order to maximize profit per unit over the course of the business period.

Ref. [**12**] presented a gated recurrent unit model based on the firefly algorithm (FA-GRU) for intelligent demand forecasting. Here the use of FA helps to select the appropriate parameter for GRU. The inaccuracies in prediction are very low compared to other methods. However, the computational time required for the model is not considered here

3. PROPOSED SYSTEM

10.48047/jocaaa.2025.34.05.41

This section gives the detailed analysis of system architecture as shown in Figure 4.1. The detailed analysis is given as follows

Step 1: Dataset Information

Begin by gathering a comprehensive dataset relevant to your time-series problem. Ensure it includes clear timestamps and associated features, such as numerical, categorical, or textual data points. Examples of such datasets include stock price data, weather data, or sensor readings. Understand the dataset's attributes, format, and structure, ensuring it has no missing timestamps or incomplete records.

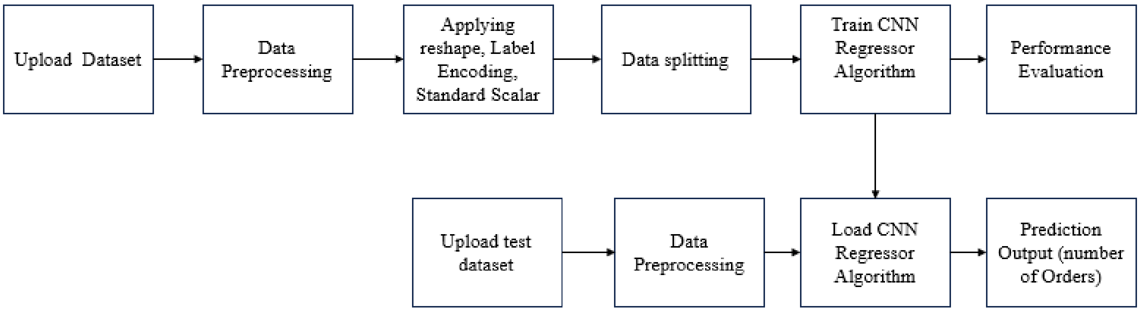

Fig. 1: Block diagram

Step 2: Dataset Preprocessing in Time Series Data

Preprocess the dataset by handling missing values, smoothing noisy data, and ensuring all timestamps are uniformly spaced. Convert the data into a time-series format by indexing with timestamps. Normalize or standardize the features to improve the convergence of algorithms. Split the dataset into training, validation, and testing sets, ensuring no data leakage between sets due to overlapping time periods.

Step 5: Existing Algorithm - XGBoost Algorithm

Implement the XGBoost algorithm as a baseline for the task. XGBoost, a robust ensemble learning method, works by creating multiple decision trees and averaging their outputs to improve accuracy and reduce overfitting. Train the model on the preprocessed dataset and evaluate its performance using metrics such as accuracy, mean absolute error (MAE), or root mean square error (RMSE) based on the task requirements.

Step 6: Proposed Algorithm - Convolutional Neural Network (CNN)

Develop and train a Convolutional Neural Network tailored for time-series data. CNNs are excellent at learning patterns from raw data through their hierarchical feature extraction capabilities. Design an architecture suited for time-series, such as including 1D convolutional layers, pooling layers, and

10.48047/jocaaa.2025.34.05.41

dense layers for final predictions. Train the CNN on the dataset and optimize its performance using hyperparameter tuning.

Step 7: Performance Comparison

Compare the performance of the XGBoost algorithm and the CNN model using evaluation metrics. Highlight improvements in accuracy, reduction in error rates, or enhanced feature extraction capabilities. Visualize the results using plots such as prediction curves, confusion matrices, or metric comparison bars to showcase the superiority of the CNN over the XGBoost for the given task.

4. RESULTS AND DESCRIPTION

Figure 2 shows the homepage of the food demand prediction application, rendered by the home view in views.py. This page serves as the entry point for users, displayed when accessing the root URL (/) as defined in urls.py (path('', views.home)). It is implemented using the Home.html template, which likely contains a user-friendly interface with navigation options to other pages such as registration, login, data upload, or prediction. The page does not process data or display metrics but provides a starting point for user interaction, potentially including links or buttons to access features like register, login, or upload. No specific values are associated with this figure, as it is a static or dynamic landing page rendered via render(request, 'Home.html').

Fig. 2: Home Page.

Figure 3 shows the registration page, rendered by the register view in views.py and accessed via the /register URL (path('register', views.register, name='register')). This page displays a form for users to input their first name, email, username, password, confirmation password, and role (admin or non-admin). The form submission triggers a POST request, where the view validates inputs (e.g., checks for unique username/email, matching passwords) and creates a user with User.objects.create_user. If validation fails, error messages like “Username already exists” or “Passwords do not match” are

10.48047/jocaaa.2025.34.05.41

displayed using Django's messages framework. On success, the user is redirected to the login page. No numerical values are displayed, but the page handles user inputs dynamically.

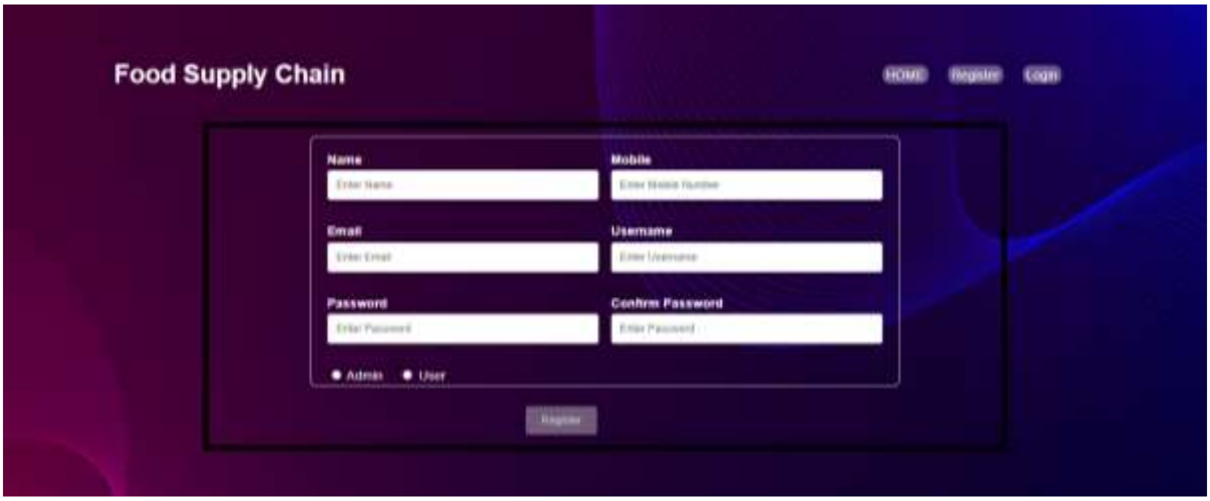

Fig. 3: Register Page

Figure 4 shows the dataset upload page, rendered by the Upload_data view in views.py and accessed via the /upload URL (path('upload', views.Upload_data, name='upload')). This page contains a form for uploading two CSV files: a primary dataset (file) and a fulfillment center dataset (center). The form submission triggers a POST request, where the files are saved temporarily using default_storage.save and processed with pandas (pd.read_csv). The view merges the datasets on center_id, preprocesses data (e.g., filling NaN with 0, encoding center_type with LabelEncoder, normalizing with MinMaxScaler), and splits data into training and testing sets. No specific values are displayed on this page, but it enables users to initiate the data processing pipeline.

Figure 5 shows the uploaded dataset preview, rendered in the prediction.html template by the Upload_data view after processing the uploaded CSV files. The view returns the first 100 rows of the processed dataset as an HTML table using outdata.to_html(), where outdata is a pandas DataFrame containing the merged and preprocessed data. The dataset includes columns like center_id, center_type, week, and num_orders, with values such as maximum base price, mean base price, minimum base price, and counts of center types (e.g., Center Type A, B, C) computed during preprocessing. For example, the view calculates max_base_price = np.max(dataset['base_price']), base_price_mean = np.mean(dataset['base_price']), and center_count for unique center types. This table provides users with a snapshot of the uploaded data.

10.48047/jocaaa.2025.34.05.41

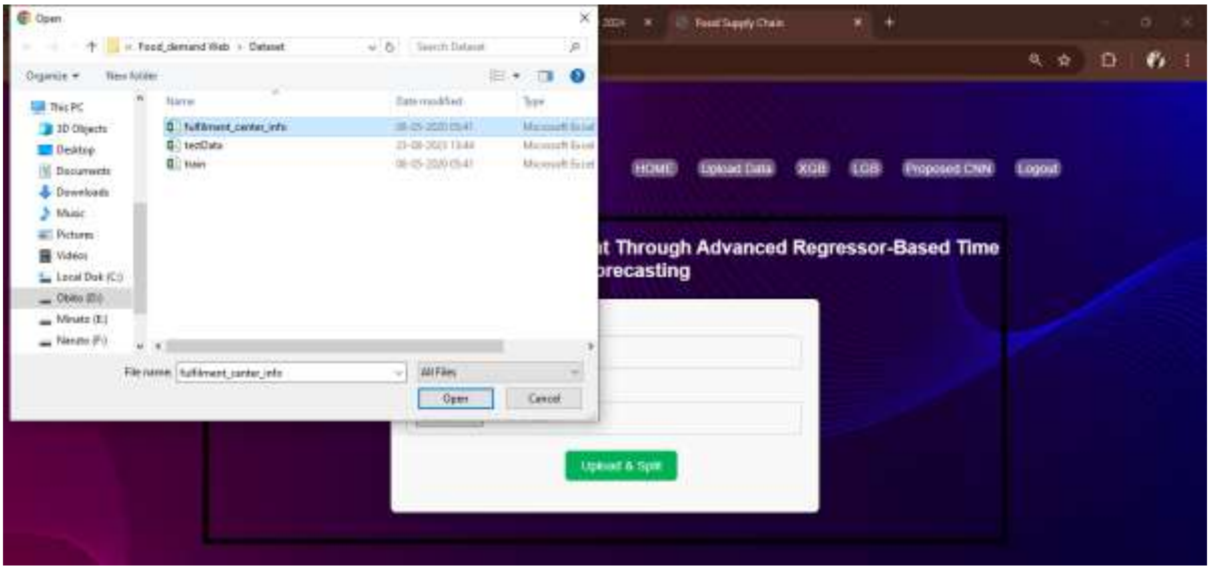

Fig. 4: Uploading Dataset

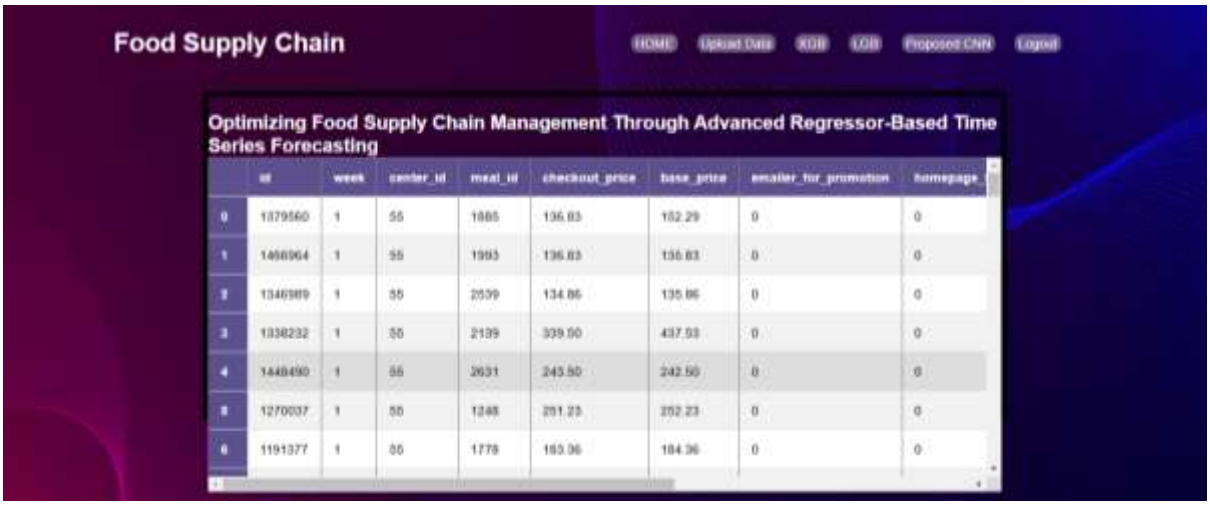

Fig. 5: Uploaded Dataset

Figure 6 shows a line plot generated by the calculateMetrics function in views.py, used in the cnn_model, lgbmodel, and xgbmodel views. The plot visualizes predicted versus actual sales, with the x-axis representing the test data indices and the y-axis showing sales values (inverse-transformed using sc2.inverse_transform). The plot includes two lines: one for actual sales (test_label, plotted in red) and one for predicted sales (predict, plotted in green). The title is set as "[Algorithm] Sales Prediction" (e.g., "Extension CNN Sales Prediction"), with labels for axes and a legend. No specific numerical values are provided in the figure description, but the plot visually compares model performance, highlighting prediction accuracy and trends.

10.48047/jocaaa.2025.34.05.41

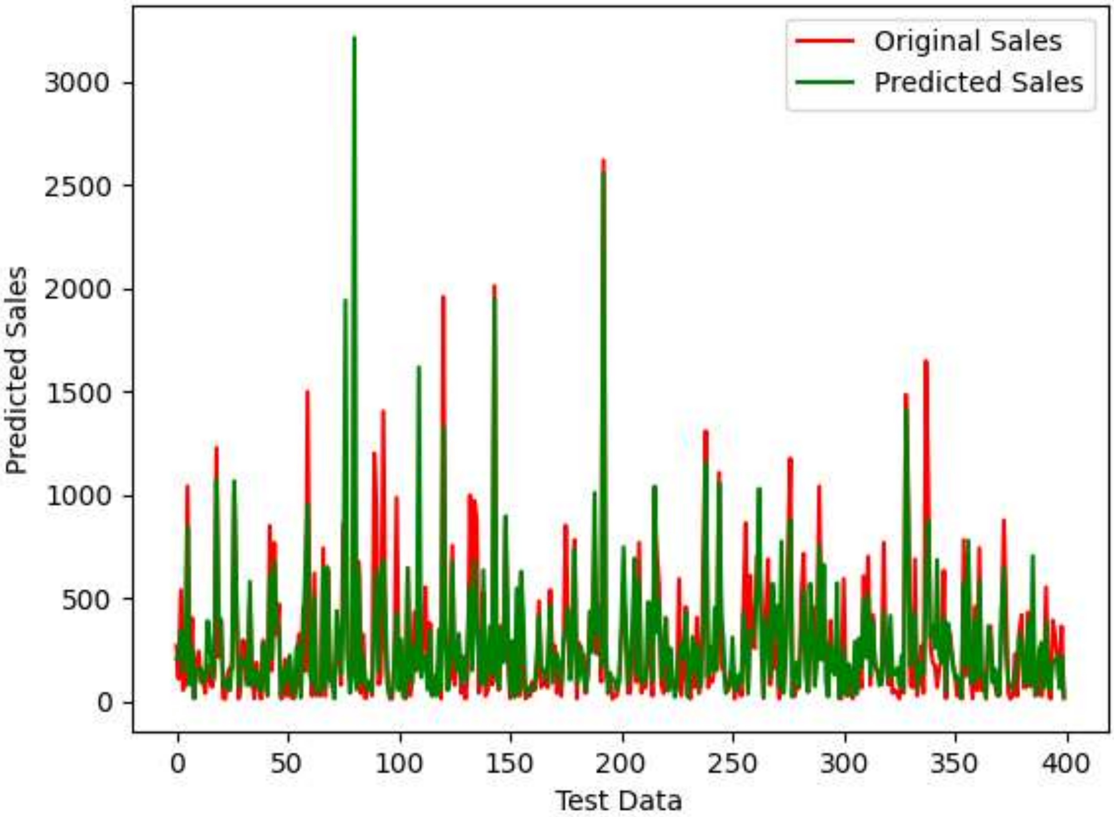

Fig. 6: Line plots.

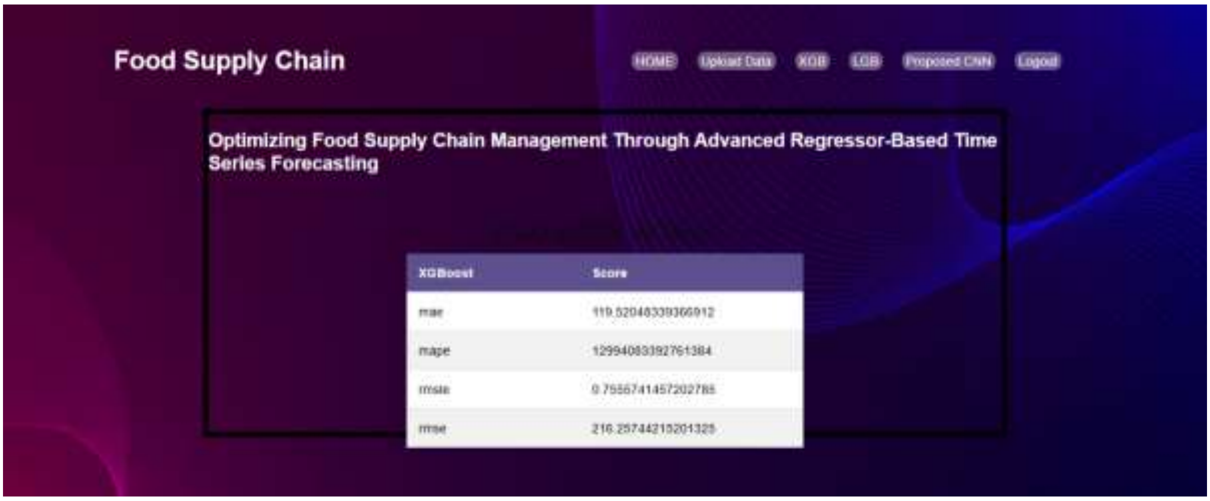

Fig. 7: Metrics of XGBoost Regressor.

Figure 7 shows the performance metrics for the XGBoost regressor, calculated by the XGB Regressor model view and displayed in prediction.html. The metrics are computed in calculate metrics and include:

- **MAE**: 119.52, indicating an average prediction error of 119.52 units, suggesting moderate accuracy but potential for improvement.

10.48047/jocaaa.2025.34.05.41

- **MAPE**: 1.2994083392761384e+13, an extremely large value, likely due to small actual values causing division issues or outliers, indicating a calculation error or data issue.
- **RMSLE**: 0.7556, showing moderate logarithmic error, suitable for datasets with skewed distributions.
- **RMSE**: 216.26, reflecting an average error of 216.26 units, sensitive to larger errors. These metrics are rendered as part of the prediction.html template, providing users with a quantitative evaluation of the XGBoost model's performance.

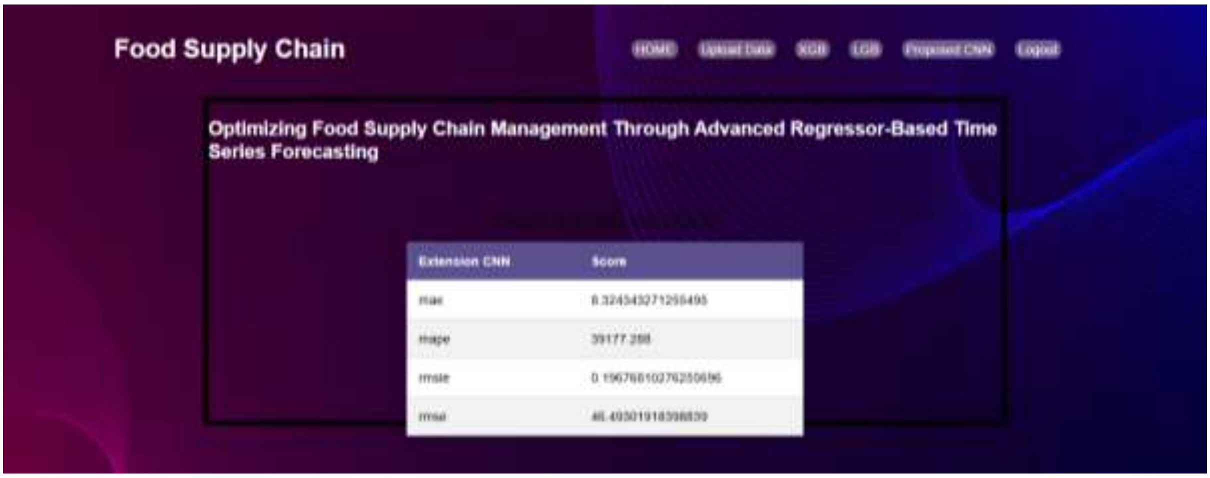

Fig. 8: Metrics of the CNN Regressor.

Figure 8 shows the performance metrics for the CNN regressor, calculated by the cnn_model view and displayed in prediction.html. The metrics are:

- **MAE**: 8.32, indicating a low average error of 8.32 units, suggesting high prediction accuracy.
- **MAPE**: 39177.29, a large value, likely due to small actual values or outliers, indicating potential issues in percentage error calculation.
- **RMSLE**: 0.1968, a low value, showing excellent performance on logarithmic error, ideal for skewed data.
- **RMSE**: 46.49, reflecting an average error of 46.49 units, moderate but better than XGBoost. These metrics are displayed in prediction.html, highlighting the CNN model's superior performance compared to XGBoost.

The comparison in table 1 summarizes the performance metrics of the XGBoost and CNN regressors, as calculated by the xgbmodel and cnn_model views in views.py and displayed in Figures 9.6 and 9.7. The MAE for XGBoost is 119.52, indicating an average prediction error of 119.52 units, while the CNN's MAE is significantly lower at 8.32, suggesting the CNN model is much more accurate on average. The MAPE for XGBoost is an extremely large 1.2994083392761384e+13, likely due to

10.48047/jocaaa.2025.34.05.41

division by near-zero actual values or outliers, rendering it unreliable, while the CNN's MAPE of 39177.29 is also large but orders of magnitude smaller, though still indicating potential data issues.

Table 1. Comparison of XGBoost and CNN Regressor Metrics

Metric	XGBoost Regressor	CNN Regressor
MAE	119.52	8.32
MAPE	1.2994083392761384e+13	39177.29
RMSLE	0.7556	0.1968
RMSE	216.26	46.49

The RMSLE for XGBoost is 0.7556, reflecting moderate logarithmic error, whereas the CNN's RMSLE of 0.1968 is much lower, demonstrating superior performance for skewed data. Finally, the RMSE for XGBoost is 216.26, indicating higher sensitivity to large errors, compared to the CNN's RMSE of 46.49, which is significantly lower, confirming better overall accuracy. Overall, the CNN regressor outperforms XGBoost across all metrics, though both models' high MAPE values suggest potential issues with small actual values or outliers in the dataset, as processed in the Upload_data and prediction views.

5. CONCLUSION AND FUTURE SCOPE

Optimizing food supply chain management through advanced regressor-based time series forecasting offers significant improvements over traditional methods. By utilizing machine learning techniques like CNNs advanced regressors, the system can predict demand more accurately and in real-time, leading to better resource allocation and inventory management. Traditional methods, such as rule-based forecasting and human intuition, often lead to inefficiencies like food wastage, stockouts, and excessive production costs. In contrast, AI-driven models help reduce these issues by considering dynamic factors like promotions, seasonality, and regional demand patterns. This system provides substantial economic benefits, especially in countries like India, where food wastage is a major issue. By aligning supply with actual demand, this approach contributes to sustainability, economic efficiency, and improved decision-making for food supply chain stakeholders.

REFERENCES

[1] Gonçalves, J.N.; Cortez, P.; Carvalho, M.S.; Frazao, N.M. A multivariate approach for multi-step demand forecasting in assembly industries: Empirical evidence from an automotive supply chain. *Decis. Support Syst.* 2021, *142*, 113452.

10.48047/jocaaa.2025.34.05.41

[2] Kilimci, Z.H.; Akyuz, A.O.; Uysal, M.; Akyokus, S.; Uysal, M.O.; Bulbul, B.A.; Ekmis, M.A. An improved demand forecasting model using deep learning approach and proposed decision integration strategy for supply chain. In *Complexity*; Wiley: Hoboken, NJ, USA, 2019.

[3] Feizabadi, J. Machine learning demand forecasting and supply chain performance. *Int. J. Logist. Res. Appl.* 2022, *25*, 119–142.

[4] Fu, W.; Chien, C.F. UNISON data-driven intermittent demand forecast framework to empower supply chain resilience and an empirical study in electronics distribution. *Comput. Ind. Eng.* 2019, *135*, 940–949.

[5] Abbasimehr, H.; Shabani, M.; Yousefi, M. An optimized model using the LSTM network for demand forecasting. *Comput. Ind. Eng.* 2020, *143*, 106435.

[6] Islam, S.; Amin, S.H. Prediction of probable backorder scenarios in the supply chain using Distributed XGBoost and Gradient Boosting Machine learning techniques. *J. Big Data* 2020, *7*, 65.

[7] Van Nguyen, T.; Zhou, L.; Chong, A.Y.L.; Li, B.; Pu, X. Predicting customer demand for remanufactured products: A data-mining approach. *Eur. J. Oper. Res.* 2020, *281*, 543–558.

[8] Leung, K.H.; Mo, D.Y.; Ho, G.T.; Wu, C.H.; Huang, G.Q. Modelling near-real-time order arrival demand in e-commerce context: A machine learning predictive methodology. *Ind. Manag. Data Syst.* 2020, *120*, 1149–1174.

[9] Saha, S.; Alrasheedi, A.F.; Khan, M.A.A.; Manna, A.K. Optimal strategies for green investment, sharing contract and advertisement effort in a supply chain coordination problem. *Ain Shams Eng. J.* 2024, *15*, 102595.

[10] Kumar Ghosh, P.; Manna, A.K.; Dey, J.K.; Kar, S. Optimal policy for an inventory system with retailer's hybrid payment strategy and supplier's price discount facility under a supply chain management. In *Optimization*; Taylor & Francis: London, UK, 2023; pp. 1–40.

[11] Dolai, M.; Manna, A.K.; Mondal, S.K. Sustainable manufacturing model with considering greenhouse gas emission and screening process of imperfect items under stochastic environment. *Int. J. Appl. Comput. Math.* 2022, *8*, 93.

[12] Al-Khazraji, H.; Nasser, A.R.; Khlil, S. An intelligent demand forecasting model using a hybrid of metaheuristic optimization and deep learning algorithm for predicting concrete block production. *IAES Int. J. Artif. Intell.* 2022, *11*, 649.