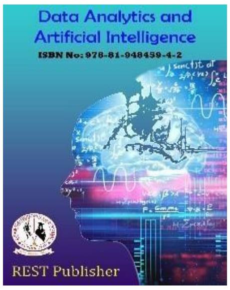
Data Analytics and Artificial Intelligence

Vol: 5(1), 2025

REST Publisher; ISBN: 978-81-948459-4-2

Website: http://restpublisher.com/book-series/daai/

DOI: https://doi.org/10.46632/daai/5/1/13



Neural Network-Based Decision Support Model for Seasonal Demand Strategy Selection

Nivetha Martin

Arul Anandar College (Autonomous), Karumathur, Madurai, Tamil Nadu, India.

Corresponding author email: nivetha.martin710@gmail.com

Abstract: *Seasonal demand fluctuations is a constraint encountered by the business personnel in inventory management and production planning. The strategies are formulated by the managerial to handle the situations of demand variations. This research work develops a neural network-based decision-making model to select the most feasible strategy to handle the seasonal demands. The proposed neural network model considers significant features such as historical demand patterns, storage costs, lead time variability, and promotional intensity, to predict the optimal strategy. The simulated data sets are used to train and validate the model. Inferences are drawn from the visualizations and classification results. This decision support system certainly facilitates the industrial decision makers to handle risks of inventory management more effectively.*

Keywords: *Neural networks, Seasonal Demand, Optimal Strategy, Inventory*

1. INTRODUCTION

Seasonal fluctuations in demand is an intricate problem faced by the organizations operating in markets. The variations in demand are a huge constraint to inventory, production, and resource allocation. The managerial evolve several strategies based on certain rules, however manual influenced rule-based decision making may not be effective in dealing with dynamic market conditions. This unveils the need of machine learning intervention particularly the neural networks. This kind of machine learning offers a more robust framework for making decisions based on the learning patterns obtained from historical and operational data.

A neural network is basically a bio inspired model, which is primarily applied in pattern recognition to make predictions and decisions. The neural network comprises of interconnected layers with associated weights. The input layers consist of the features, the hidden layers process through the activation function and the output layer produces the final results. The neural networks are efficient in handling the non-linear relationships and are widely applied in the areas of image recognition, speech processing, and decision-making systems. Neural networks are applied in industrial settings especially to predict demands. Some of the recent applications of neural network are, Zhang et al. [12] applied multi-frequency spatial-temporal graph neural network in short-term metro OD demand prediction. Fouchal et al. [4] employed artificial neural network and random forest models in biological oxygen demand estimation. Li et al. [8] used multi-scale one-dimensional convolutional neural network in predicting water chemical oxygen demand. Li et al. [9] applied graph neural network in tourism demand forecasting. Zhang [13] discussed hybrid temporal neural network for a sustainable tourism. Fallah et al. [3] estimated the demand of medical tourists using artificial neural networks.

Wu et al. [11] employed PSO trained quantile regression neural network in spare part demand forecasting. Alsamraee and Khanna [1] used advanced deep learning artificial neural network algorithms in long term electricity demand forecasting. Liu et al. [10] applied neural network approach in load demand forecasting. Irhuma et al. [7] employed convolutional neural network for supply chain demand forecasting. Zhou et al. [14] applied attention-based deep neural network in forecasting the demand of electric vehicle charging stations. Zhu et al. [15] utilized deep neural network for product demand forecasting. Chaudhary et al. [2] applied autoregressive integrated moving average and graph neural network for supply chain optimization in forecasting retail sales demand. Gong et al. [5] discussed energy demand forecasting and economic regulation system optimization based on neural network. Honcharenko et al. [6] applied artificial neural network technologies in food market demand forecasting.The above stated applications of neural network primarily focus on demand forecasting. However, the review of literature does not carry any information on neural networks addressing seasonal demands. This has motivated the authors to evolve a neural network model to handle the situation of seasonal demand. The primary objective of this model is not to make any predictions or forecasting, rather this model is classification kind. The architecture of this neural network is designed to classify outputs into three categories of strategies for managing seasonal demand.

The remaining contents of the paper are structured as follows, section 2 discusses the methodology of neural network modelling with the problem of strategic choice making. Section 3 presents the coding of proposed classification model. Section 4 consists of the results and the last section concludes the work.

2. METHODOLOGY OF NEURAL NETWORK

This section presents the steps involved in the proposed decision-making model based on neural network.

Problem Definition

Initially the problem is well defined with the selection of both input and output features. The classification model is well determined with the input features and output labels.
For instance, let us consider a company experiences seasonal demand fluctuations and adopts one of three strategies to handle them:

- S_1: Increase inventory buffer
- S_2: Flexible workforce allocation
- S_3: Dynamic pricing adjustments

The goal is to predict which strategy is optimal given historical and real-time business features using a Neural Network (NN).

A three class classification model is formulated with six input features and 3 output labels.
The input features are as follows in Table 1.

TABLE 1. Input Features

Feature Name	**Description**
X1	Seasonality indicator (0 = off-season, 1 = peak season)
X2	Average weekly demand (units/week)
X3	Storage cost per unit
X4	Supplier lead time (in days)
X5	Coefficient of variation of demand
X6	Promotion intensity (0 = no promotion, 1 = heavy promotion)

Generation of Data

A hypothetical data set comprising 1000 samples is generated
X=[SeasonIndex, AvgWeeklyDemand, StorageCostPerUnit, LeadTimeDays, DemandVariability, PromotionIntensity]
Each feature is sampled from simple distributions (uniform or integer ranges) to mimic realistic variation.
Rule based Decision

A deterministic rule assigns class $y \in \{1,2,3\}$ for each sample:

- If Season Index > 0.7 and Lead Time Days > 15 → y=1 (S1).
- Else if Demand Variability <0.3 and Storage Cost Per Unit < 1.5 → y = 2 (S2).
- Otherwise → y=3 (S3).

This produces labelled training data $\{(x_i,y_i)\}^{N}_{i=1}$

Data Preprocessing

- Features matrix $X \in R^{N \times d}$, target encoded as one-hot matrix $Y \in \{0,1\}^{N \times 3}$
- compute mean μ_j and standard deviation σ_j on training set, transform each numeric feature using

$$\widetilde{x_{ij}} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

- Split the training and test sets via random split (80/20).

Model Architecture (Neural Network)

A feed-forward neural network (Keras Sequential):

- Input: d standardized features.
- Hidden layer 1: 16 neurons, ReLU activation.
- Hidden layer 2: 8 neurons, ReLU activation.
- Output layer: 3 neurons, Softmax activation to produce class probabilities.

Mathematically, for input $\tilde{x}$

$h^{(1)} = \mathrm{ReLU}(W^{(1)} \ \tilde{x} + b^{(1)})$

$h^{(2)} = \mathrm{ReLU}(W^{(2)} \ \tilde{x} + b^{(2)})$

$z = W^{(3)} h^{(2)} + b^{(3)}$

The softmax output is

$$\hat{y}_k = \frac{e^{z_k}}{\sum_{m=1}^{3} e^{z_m}}$$

Categorical cross-entropy is used to calculate the loss of the model. The training is performed and on using the evaluation metrics, the accuracy and the performance of the model is determined.

3. PYTHON CODING FOR NEURAL NETWORK MODELLING

```
# --------------------------
# Seasonal Demand Strategy Prediction with Neural Network
# --------------------------
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# 1. Simulate Dataset
np.random.seed(42)
n_samples = 1000

# Features
season_index = np.random.uniform(0, 1, n_samples)        # 0 = off-season, 1 = peak season
avg_weekly_demand = np.random.randint(50, 500, n_samples)  # units/week
storage_cost_per_unit = np.random.uniform(0.5, 3.0, n_samples) # cost per unit
lead_time_days = np.random.randint(1, 30, n_samples)       # supplier lead time
demand_variability = np.random.uniform(0, 1, n_samples)    # coefficient of variation
promotion_intensity = np.random.uniform(0, 1, n_samples)   # 0 = no promo, 1 = heavy promo

# Strategy assignment logic (synthetic rule for simulation)
# Strategy 1: Preseason stockpile (high season index, long lead time)
```

```
# Strategy 2: Just-in-time ordering (low variability, low storage cost)
# Strategy 3: Mixed approach (moderate conditions)
strategy = []
for i in range(n_samples):
    if season_index[i] > 0.7 and lead_time_days[i] > 15:
        strategy.append(1)
    elif demand_variability[i] < 0.3 and storage_cost_per_unit[i] < 1.5:
        strategy.append(2)
    else:
        strategy.append(3)

# Create DataFrame
df = pd.DataFrame({
    'SeasonIndex': season_index,
    'AvgWeeklyDemand': avg_weekly_demand,
    'StorageCostPerUnit': storage_cost_per_unit,
    'LeadTimeDays': lead_time_days,
    'DemandVariability': demand_variability,
    'PromotionIntensity': promotion_intensity,
    'Strategy': strategy
})
print(df.head())

# 2. Data Preprocessing
X = df.drop('Strategy', axis=1)
y = pd.get_dummies(df['Strategy'])  # one-hot encoding

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# 3. Neural Network Model
model = Sequential([
    Dense(16, activation='relu', input_shape=(X_train.shape[1],)),
    Dense(8, activation='relu'),
    Dense(3, activation='softmax')  # 3 strategies
])
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# 4. Train the Model
history = model.fit(X_train, y_train, epochs=50, batch_size=16, validation_split=0.2, verbose=0)

# 5. Model Evaluation
loss, accuracy = model.evaluate(X_test, y_test, verbose=0)
print(f"Test Accuracy: {accuracy:.2f}")

# 6. Predictions
y_pred = model.predict(X_test)
y_pred_classes = np.argmax(y_pred, axis=1)
```

```
y_true_classes = np.argmax(y_test.values, axis=1)

# 7. Classification Report & Confusion Matrix
print("\nClassification Report:\n", classification_report(y_true_classes, y_pred_classes))

cm = confusion_matrix(y_true_classes, y_pred_classes)
plt.figure(figsize=(6, 5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['S1', 'S2', 'S3'], yticklabels=['S1', 'S2', 'S3'])
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()

# 8. Feature Distribution Plot
df_melt = df.melt(id_vars='Strategy', var_name='Feature', value_name='Value')
plt.figure(figsize=(10, 6))
sns.boxplot(x='Feature', y='Value', hue='Strategy', data=df_melt)
plt.xticks(rotation=45)
plt.title('Feature Distributions by Strategy')
plt.show()

# 9. Training Curves
plt.figure(figsize=(10, 4))
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.title('Accuracy Curve')

plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.title('Loss Curve')
plt.show()
```

4. RESULTS AND DISCUSSION

The classification report is presented in Table 2.

TABLE 2. Classification Report

Class	Precision	Recall	F1-Score	Support
Strategy 1	0.91	0.95	0.93	225
Strategy 2	0.83	0.83	0.83	130
Strategy 3	0.95	0.93	0.94	645
Accuracy			0.92	1000
Macro Avg	0.90	0.90	0.90	1000
Weighted Avg	0.92	0.92	0.92	1000

The classification model achieved an overall accuracy of 92% across 1,000 samples, demonstrating strong predictive performance. Strategy 1 showed high precision (0.91) and recall (0.95), resulting in an F1-score of 0.93. Strategy 2 had slightly lower but balanced precision and recall at 0.83 each, yielding an F1-score of 0.83. Strategy 3 performed best overall with a precision of 0.95, recall of 0.93, and an F1-score of 0.94. The macro averages for precision, recall, and F1-score were all 0.90, while the weighted averages matched the accuracy at 0.92, indicating consistent performance across classes despite differences in class support.

The boxplot in Fig 1 illustrates the distribution of key features across the three seasonal demand strategies. Strategy 1 generally operates in higher seasonal index ranges and handles larger average weekly demand, often with longer lead times, suggesting a stockpiling-oriented approach. Strategy 2 is positioned at the lower end for seasonal index, average weekly demand, lead time, and demand variability, reflecting its just-in-time ordering nature with minimal inventory holding. Strategy 3 typically falls between the other two in seasonal index and demand, showing moderate values across most features. Storage cost per unit appears relatively consistent across strategies, though Strategy 2 leans slightly toward the lower end. Promotion intensity shows overlapping ranges for all strategies, indicating that promotional activities are not the primary differentiator among them.

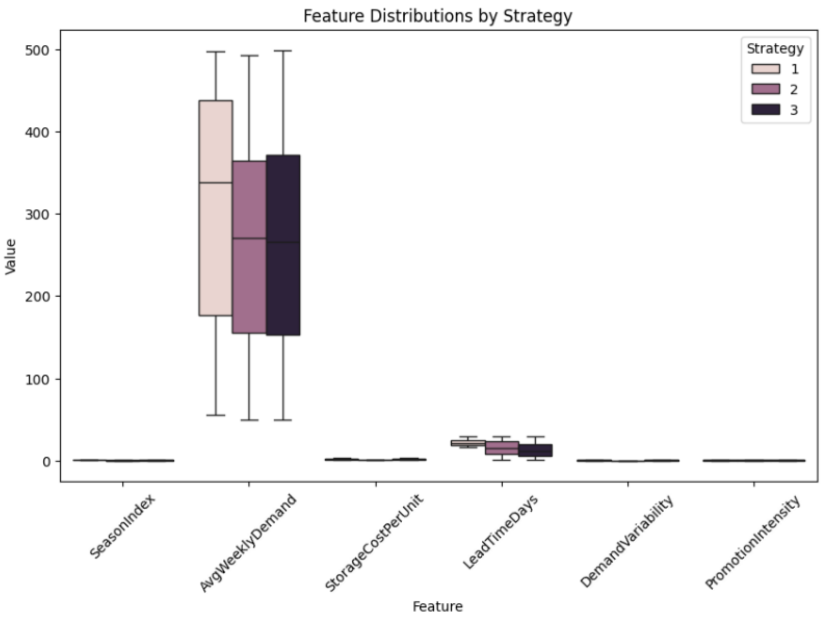


FIGURE 2. Boxplot of Feature Distribution by Strategy

The training and validation curves in Fig. 2 indicate that the neural network learns effectively during the early epochs, with both accuracy improving and loss decreasing up to around epoch 20–25. After this point, the training accuracy continues to rise toward about 95%, and training loss keeps decreasing.

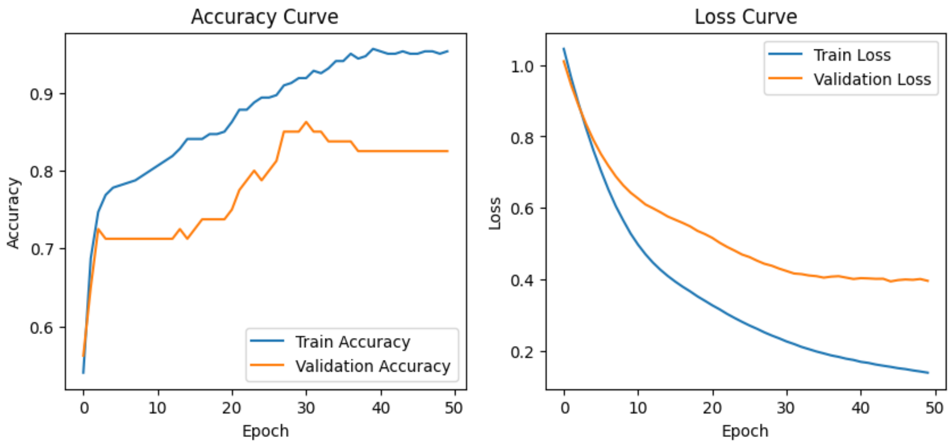


FIGURE 2. Accuracy and Training Curves

5. CONCLUSION

The proposed neural network-based decision support model offers significant industrial implications by enabling businesses to effectively address seasonal demand fluctuations through data-driven strategy selection. By integrating key factors such as historical demand patterns, storage costs, lead time variability, and promotional intensity, the model assists managers in making timely and accurate decisions, thereby minimizing stockouts, overstocking, and associated costs. The classification results, with an overall accuracy of 92%, demonstrate the model's reliability in predicting the most suitable strategy. This approach not only enhances operational efficiency but also strengthens competitive advantage by ensuring optimal inventory management. In conclusion, the model serves as a practical and robust tool for industrial decision-makers, promoting cost savings, improved resource utilization, and resilience against market uncertainties.

REFERENCES

[1]. Alsamraee, S. A., & Khanna, S. (2025). Long-Term Electricity Demand Forecasting of a University Campus Based on Advanced Deep Learning Artificial Neural Network Algorithms. *Results in Engineering*, 106683.

[2]. Chaudhary, A., Husayn, S. O., Vallabhaneni, R., Yarram, S., & KS, S. (2025, May). Forecasting Retail Sales Demand by using AutoRegressive Integrated Moving Average and Graph Neural Network for Supply Chain Optimization. In *2025 3rd International Conference on Data Science and Information System (ICDSIS)* (pp. 1-6). IEEE.

[3]. Fallah Tafti, H., Esteghlal, A., Al-Modaresi, S. A., Beheshtipour, Z., & Mirhosseini, S. M. (2025). Forecasting the Demand of Medical Tourists in Yazd Using Artificial Neural Network. *Management Strategies in Health System*, *9*(4), 314-328.

[4]. Fouchal, A., Tikhamarine, Y., Benbouras, M. A., Souag-Gamane, D., & Heddam, S. (2025). Biological oxygen demand prediction using artificial neural network and random forest models enhanced by the neural architecture search algorithm. *Modeling Earth Systems and Environment*, *11*(1), 9.

[5]. Gong, X., Peng, B., Li, Y., & Wang, Z. (2025, March). Energy Demand Forecasting and Economic Regulation System Optimization Based on Neural Network. In *2025 IEEE International Conference on Electronics, Energy Systems and Power Engineering (EESPE)* (pp. 01-05). IEEE.

[6]. Honcharenko-Halitsyn, S., Kopp, A. M., Halatova, O. S., & Martynyuk, S. (2025). Research on the Ukrainian food market demand forecasting using artificial neural network technologies.

[7]. Irhuma, M., Alzubi, A., Öz, T., & Iyiola, K. (2025). Migrative armadillo optimization enabled a one-dimensional quantum convolutional neural network for supply chain demand forecasting. *PloS one*, *20*(3), e0318851.

[8]. Li, J., Lu, Y., Ding, Y., Zhou, C., Liu, J., Shao, Z., & Nian, Y. (2025). Prediction of Water Chemical Oxygen Demand with Multi-Scale One-Dimensional Convolutional Neural Network Fusion and Ultraviolet–Visible Spectroscopy. *Biomimetics*, *10*(3), 191.

[9]. Liang, X., Li, X., Shu, L., Wang, X., & Luo, P. (2025). Tourism demand forecasting using graph neural network. *Current Issues in Tourism*, *28*(6), 982-1001.

[10]. Liu, M., Zhao, W., Zhou, Y., & Eslami, M. (2025). Load demand forecasting in air conditioning a rotor Hopfield neural network approach optimized by a new optimization algorithm. *Scientific Reports*, *15*(1), 18774.

[11]. Wu, C. G., Fu, X., & Xia, Y. (2025). Spare part demand forecasting using PSO trained Quantile Regression Neural Network. *Computers & Industrial Engineering*, *200*, 110841.

[12]. Zhang, J., Zhang, S., Zhao, H., Yang, Y., & Liang, M. (2025). Multi-frequency spatial-temporal graph neural network for short-term metro OD demand prediction during public health emergencies. *Transportation*, 1-23.

[13]. Zhang, Y., Tan, W. H., & Zeng, Z. (2025). Tourism Demand Forecasting Based on a Hybrid Temporal Neural Network Model for Sustainable Tourism. *Sustainability*, *17*(5), 2210.

[14]. Zhou, Z., Jiang, H., Yao, L., Wang, S., Che, H., & Gu, Y. (2025). Multi-Demand Forecasting for Electric Vehicle Charging Stations under Time-of-Use Strategy via Attention-Based Deep Neural Network. *IEEE Internet of Things Journal*.

[15]. Zhu, H., Yang, S., Gupta, T., Waldon, H., Olivares, K. G., & Gopalsamy, R. (2025). RSight: A deep neural network for product demand forecasting over geographic regions.