International Journal of Computational and Experimental Science and ENgineering
(IJCESEN)
Vol. 10-No.4 (2024) pp. 811-826
http://www.ijcesen.com
ISSN: 2149-9144

Research Article

Assessing the Profit Impact of ARIMA and Neural Network Demand Forecasts in Retail Inventory Replenishment

A. Burak PAÇ[1]*, Betül YAKUT[2]

[1]Gebze Technical University, Faculty of Engineering, Industrial Engineering Department, 41400, Kocaeli-Turkiye
* **Corresponding Author Email:** burakpac@gtu.edu.tr - **ORCID:** 0000-0003-2177-4585

[2]Gebze Technical University, Faculty of Engineering, Industrial Engineering Department, 41400, Kocaeli-Turkiye
Email: b.yakut2021@gtu.edu.tr - **ORCID:** 0009-0009-8866-1535

Article Info:

DOI: 10.22399/ijcesen.439
Received : 02 September 2024
Accepted : 30 October 2024

Keywords:

Inventory Replenishment
Sales Demand Forecasting
ARIMA
Artificial Neural Networks
Integer Programming

Abstract:

This study explores the integration of demand forecasting and inventory replenishment strategies to enhance retail profitability. Accurate sales forecasting is essential for efficient inventory replenishment decisions. Both traditional ARIMA and modern neural network models are utilized to predict future sales. These forecasts input into an integer programming model that strategically manages the inventory of stores across multiple retail routes. The optimization model considers transportation, sales loss, supply costs, and inventory dynamics to maximize retail profit with daily replenishment decisions. This approach enables us to assess the impact of forecasting accuracy on profitability over a multi-period planning horizon. The study is distinctive in its dual assessment: it evaluates both the accuracy of forecasting methods and their direct impact on profitability through systematic inventory decisions. Neural network architectures exhibit a 6% lower mean squared error compared to ARIMA models. For longer horizon predictions, the performance gap grows larger; for example, there is a 60% difference in predictions 15 days ahead. Predictions reflect 1.6% higher profits on average when neural network predictions and more efficient longer planning horizons of the optimization model are preferred. Planning 30 days ahead, optimizing with neural network predictions elicits 2.3% higher profits compared to those attainable based on ARIMA predictions. Our findings illustrate how different forecasting methods can affect firm profitability by shaping inventory replenishment strategies. By merging mathematical optimization with time series forecasting, this research provides a comprehensive evaluation of how advanced predictive technologies can enhance retail inventory practices and improve profitability.

1. Introduction

Inventory management is critically strategic for businesses aiming to optimize operations amidst fierce market competition. In industries with limited profit margins, companies must focus on optimizing resource management to maintain competitiveness and profitability. A substantial portion of a firm's capital is often tied to working capital, with inventories usually representing the largest share. Effective inventory management, therefore, is synonymous with sound financial management.

Efficient capital utilization not only improves customer service but also enhances profitability. Achieving these goals involves maximizing output and profit margins with available capital, necessitating rapid capital turnover—an often challenging goal. Moreover, the overarching financial objective of maximizing firm value should guide inventory management systems. Traditional financial models primarily focus on the challenging task of maximizing net profits [1]; while increasingly adopting value maximization strategies.

The primary financial objective of any firm is to maximize its value, and effective inventory management plays a crucial role in achieving this goal. Traditional asset management models often focus on maximizing book profit, yet enhancing firm value encompasses more than just profit figures; it involves careful management of all current assets including receivables, inventories, and cash balances.

In inventory management, decision-making involves a critical trade-off: balancing the risk of lost sales and reputation associated with low inventory levels against the costs associated. This balance is pivotal in corporate financial management, where the aim is to maintain inventory at a cost-effective minimum. High inventory levels tie up capital and incur costs related to storage, insurance, transportation, obsolescence, waste, and spoilage. Conversely, too little inventory can disrupt fulfilling demands effectively [2].

Inventory management is a dynamic process that encompasses the continuous flow of ordering, storing, producing, selling, and restocking goods [3]. Strategically, inventory serves to buffer against supply-demand mismatches, safeguard against supply failures, and minimize supply chain costs. As a managerial control mechanism, inventory management regulates the flow of goods from ordering to restocking, balancing inventory levels with market demand to minimize overall costs [4]. Such management is crucial since inventory ties up capital and incurs carrying costs, underscoring the need for meticulous planning and execution of when and how much to order and how much product to keep in stock via inventory strategies [5].

This study is designed to enhance the profitability of a retail company by optimizing its shipment processes from the central warehouse. To achieve this, it employs demand forecasting methods to predict product sales at retail outlets and develops an integer programming model aimed at maintaining store inventories at optimal levels. This model optimizes daily replenishment decisions across various routes that may encompass one or more stores. It seamlessly integrates sales demand forecasting with inventory replenishment operations, thus ensuring precise sales forecasts and optimal inventory management. This integration considers multiple factors including sales revenue and costs related to supply, inventory holding, sales loss, and transportation. Furthermore, the study establishes a comprehensive framework for demand forecasting and inventory control. Within this framework, traditional statistical time series forecasting using univariate Autoregressive Integrated Moving Average (ARIMA) models is juxtaposed against a suite of univariate neural network architectures. This dual approach facilitates a two-pronged comparison: firstly, a direct evaluation of the forecasting accuracy of different methods, and secondly, an analysis of how these forecasts impact firm profitability via the inventory replenishment strategies implemented through the optimization model. This methodology not only highlights the predictive capabilities of each forecasting technique but also underscores their practical implications on operational efficiency and profit maximization.

The organization of the paper is as follows. Chapter 2 presents the related literature; Chapter 3 discusses the optimal inventory replenishment problem and its integer programming model. Chapter 4 describes the time series modeling approach and model selection via prediction performance. Chapter 5 presents the computational results of optimization model runs with the demand parameters predicted via the selected ARIMA and neural network time series models. Chapter 6 concludes the study.

2. Literature Review

Recent advancements in inventory management have significantly contributed to optimization strategies across various industrial sectors. Specific challenges in inventory systems have led to the development of tailored strategies for managing different types of products. Systems for managing non-perishable products that integrate promotional strategies and quality control measures have been designed to optimize inventory [6]. Models accounting for the deterioration rates of goods have been developed, crucial for adapting replenishment policies to the characteristics of the products [7]. Simulation models for spare parts inventory have been created, demonstrating significant cost and space savings [8]. Additionally, dynamic lot-sizing problems have been addressed with new heuristics that prove effective in finding optimal solutions within complex inventory scenarios [9].

Theoretical advancements in inventory management have also been significant. Two-warehouse systems under variable demand rates have been examined, focusing on the economic impacts of storage capacity limitations [10], and accelerated production and reprocessing within multi-item stock systems have been investigated, providing detailed models that guide operational decisions [11]. A multi-depot, multi-item model with non-instantaneous products has been evaluated using an optimization algorithm to maximize profitability [12]. Similarly, a multi-item economic order quantity (EOQ) model under a vendor-managed inventory system that incorporates warehouse capacity and delivery constraints has been developed, applying ant colony optimization and genetic algorithms to address uncertainties in demand and storage, thereby minimizing the total cost of the supply chain [13].

Moreover, the extension of these models to diverse industrial contexts illustrates their broad applicability and critical importance. Inventory management in the agricultural sector has been addressed to mitigate post-harvest losses, utilizing a multi-item specific model to calculate the EOQ for different rice types in India [14]. Inventory-manufacturing scenarios in high-precision production systems have been explored, focusing on cost minimization and operational efficiency [15].

In exploring the landscape of sales forecasting, numerous studies have compared traditional time series analysis methods to modern artificial neural networks (ANNs). It has been underscored that ANNs demonstrate superior ability to capture dynamic, nonlinear trends and seasonal patterns over traditional methods such as Winters exponential smoothing, Box-Jenkins ARIMA model, and multivariate regression [16]. Additionally, the inadequacies of conventional forecasting techniques in apparel retail, particularly where variables like promotions and weather significantly affect demand, have been discussed. Research utilizing ANNs to forecast sales for a chain of stores in Türkiye has shown that incorporating diverse data inputs beyond historical sales aligns with a more nuanced approach to demand prediction [17].

The adaptability and customization possible with neural networks in regional and industry-specific applications are highlighted, with a focus on the footwear industry revealing the potential of neural networks to accommodate varied market dynamics [18] and enhance the precision of sales forecasts [19]. In the fashion retail sector, differences in performance between deep learning and methods such as Random Forest, Decision Trees, and Support Vector Regression have been explored. Findings suggest that while deep learning is effective, it does not always significantly outperform traditional methods, pointing to the need for contextually optimized deployment of ANNs in sales forecasting [20].

Integrating forecasting methods with inventory management decisions remains challenging. A study on fast-moving consumer goods employes machine learning algorithms, including ANNs and Binary Decision Trees, to predict EOQ and assess their impact on key performance indicators, highlighting substantial improvements in "available to promise" and "operating cash flow" [21]. Similarly, a novel decision integration strategy that blends deep learning, support vector regression, and traditional time series models has demonstrated significant accuracy improvements in demand forecasting for a major retail chain in Türkiye [22]. Both studies underscore the potential of advanced forecasting techniques to not only predict sales but also significantly enhance inventory management and profitability.

Explorations into the realm of vendor-managed inventory contracts have utilized various neural network models for demand forecasting. These studies compare the results of neural network-based forecasts on multiple fronts and employ multi-criteria decision-making tools to verify these results. The objective is to achieve substantial savings in inventory costs, thereby demonstrating the direct impact of improved forecasting on inventory management and cost reduction [23]. However, despite these advancements, a notable gap remains in comprehensive studies that assess how these forecasting improvements affect profitability through systematic replenishment decisions. This gap highlights a crucial area for future research, underscoring the need for explicit inventory and replenishment models that employ advanced forecasting techniques and critically evaluate their impact on firm profitability, potentially revolutionizing inventory management practices to align more closely with cutting-edge predictive analytics.

These contributions underscore the evolution of inventory management and demand forecasting through the integration of advanced mathematical models and innovative strategies across various sectors. This foundation supports the aim to refine these strategies further by integrating them into a comprehensive model that optimizes inventory replenishment in retail settings, enhancing profitability and operational efficiency.

This study distinctively assesses the predictive performance of various neural network architectures alongside selected ARIMA models using real retail sales data. It explores their impact on optimal firm profits through a deterministic optimal replenishment model over a multi-period planning horizon. The research employs both ARIMA and neural network models for demand forecasting, aiming to enhance operational efficiency and optimize inventory management by augmenting the devised integer programming model with the best prediction model selection. By merging mathematical optimization with time series analysis, the approach not only refines the selection of prediction models but also reinforces the accuracy and reliability of the outcomes.

Fig. 1 outlines the process of predicting future sales and optimizing inventory replenishment decisions based on historical sales data from store locations. The process begins with the collection of historical sales data, crucial for training and testing prediction models. Typically, the last 10-20% of the data is reserved for testing, while the earlier portion is used for training.

Various predictive modeling techniques can be employed to forecast future sales from this historical data. These techniques include traditional statistical time series methods such as SARIMA (Seasonal ARIMA) and exponential smoothing, as well as contemporary machine learning models like support vector machines and gradient boosting machines. Among all candidate methods, the most accurate predictors can be selected for predicting the sales based on their performance on test data. This analysis focuses on two specific classes of predictive models: ARIMA and neural networks. These models are compared not only in terms of their prediction accuracy but also in how effectively their sales forecasts can inform and improve decisions made by the integer programming optimization model for inventory replenishment.

For each store and product combination, the best-performing ARIMA model and neural network architecture are identified and selected. The evaluation of the profitability of optimized inventory decisions can be conducted in a live setting, involving daily updates to the historical data, running the optimization model, implementing the replenishment decisions prescribed by the model for the next day, and then observing and recording the resulting demand, sales, and inventory levels. Alternatively, a final portion of historical data can be reserved to simulate the prediction-optimization framework's profitability against actual daily demand, as applied here.

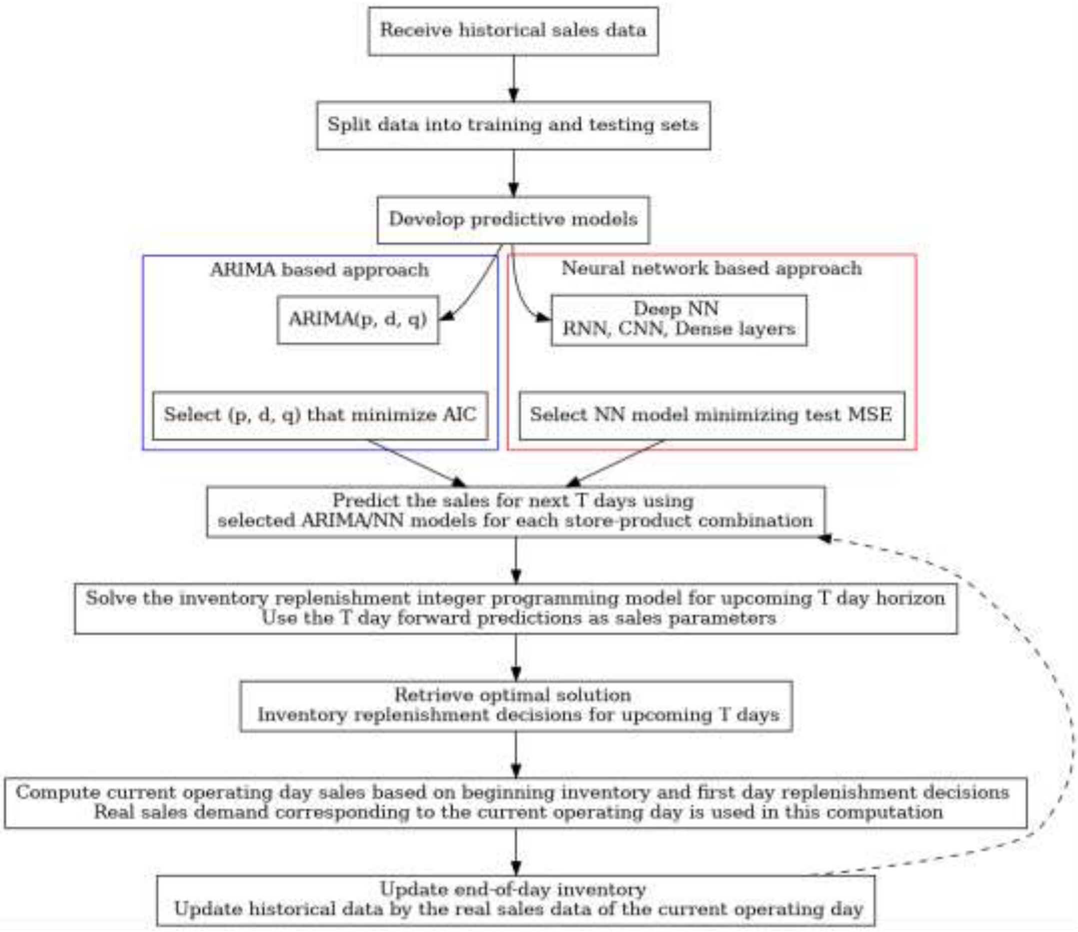

Figure 1. *Workflow of the framework: sales forecasting model selection, sales forecasting, inventory replenishment optimization, testing optimal decisions and forecast models based on real sales data.*

3. The Inventory Replenishment Integer Programming Model

Here, we devise an optimization model for defining the optimal replenishment strategy for a retail company. The company operates with one central warehouse that distributes products to several stores, each carrying the same set of product brands and models. Daily shipments can be dispatched to these stores via various routes, each with a specific fixed cost. Each route serves one or more stores.

The model considers both cost and revenue parameters that influence the decision-making process. These include the price of items on the revenue side, their supply costs, daily charges per inventory held, and a penalty for lost sales when demand exists but no inventory is available. These factors can vary daily or be specific to each store where applicable. The company aims to plan these replenishments under the anticipation of sales demand over a specified planning horizon, typically spanning several days or weeks.

The company's objective is to maximize profit, which is defined as sales revenue minus the costs of product supply, transportation for replenishment, inventory holding, and lost sales. This model aims to determine daily whether to dispatch from any of the routes and if so, how to allocate delivery capacity to stores serviced by those routes, and the set of products. The goal is to ensure that the strategy maximizes profit by efficiently balancing revenue against the combined costs of replenishments, inventory holding, and lost sales opportunities.

The indices and index sets used are as follows:

(i) $i \in \{1,\dots,N\}$: the product model indices,
(ii) $t \in \{1,\dots,T\}$: the days in the planning horizon,
(iii) $m \in \{1,\dots,M\}$: the stores in the scope of planning,
(iv) $r \in \{1,\dots,R\}$: routes that cover one or more stores for delivery,
(v) $M_r \subseteq \{1,\dots,M\}$: The set of stores covered by delivery route r,
(vi) $R_m \subseteq \{1,\dots,R\}$: The set of routes delivering to store m.

The decision variables are:

(i) x_{tim}: The number of products of type i delivered to store m on day t,
(ii) y_{tr} : Indicates whether delivery from route r occurs on day t,
(iii) v_{tmr}: Capacity allocated to store m from delivery via route r on day t,
(iv) S_{tim}: Number of products i sold at store m on day t,
(v) L_{tim}: Number of product i sales lost due to stockout at store m on day t,
(vi) I_{tim}: Product i inventory at the store m at the end of day t.

Parameters of the problem are:

(i) d_{tim}: Forecasted product i sales quantity in number of items at store m on day t,
(ii) p_{tim}: Sales price for a unit of product i at store m on day t,
(iii) f_{tr}: Fixed cost of delivery through route r on day t,
(iv) c_{ti}: Supply cost of a unit of product i on day t,
(v) B: Capacity of delivery truck in number of items,
(vi) h_{tim}: Daily inventory holding cost per item for product i at store m on day t,
(vii) q_{tim}: The cost per item of lost sales for product i at store m on day t.

These parameters provide the flexibility of setting different cost and revenue for different product, day and stores as applicable to the respective parameters, however, in reality a restricted part of this flexibility is often utilized. For instance, the cost of supplying some items can be fixed due to a contract that covers a season, thus possibly the entire planning horizon. Then, $c_{ti} = c_i$ would be fixed for that product i. Similarly, sales prices and lost sales costs can be dependent or irrespective of the specific day or store; lost sales or holding costs can further be product independent in some settings. With the index, variable and parameters defined, the proposed integer programming model is as follows. The objective function is:

$$\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{m=1}^{M} p_{tim} \cdot S_{tim} - \sum_{t=1}^{T}\sum_{r=1}^{R} f_{tr} \cdot y_{tr}$$
$$- \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{m=1}^{M} c_{ti} \cdot x_{tim}$$
$$- \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{m=1}^{M} h_{tim} \cdot I_{tim}$$

$$-\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{m=1}^{M} q_{tim}\cdot L_{tim}, \quad (1)$$

maximized under constraints:

$$\sum_{i=1}^{n} x_{tim} \le \sum_{r\in R_m} v_{tmr} \quad t=1,\cdots,T;\ m=1,\dots,M \quad (2)$$
$$\sum_{m\in M_r} v_{tmr} \le B\cdot y_{tr} \quad t=1,\cdots,T;\ r=1,\dots,R \quad (3)$$
$$I_{0im}=0 \quad i=1,\cdots,N;\quad m=1,\dots,M \quad (4)$$
$$I_{(t-1)im}+x_{tim}-S_{tim}=I_{tim} \quad t=1,\dots,T;\ i=1,\dots,N; m=1,\dots,M \quad (5)$$
$$S_{tim}+L_{tim}=d_{tim} \quad t=1,\dots,T; i=1,\dots,N; m=1,\dots,M \quad (6)$$
$$x_{tim}\in Z_+ \quad t=1,\cdots,T;\ i=1,\dots,N;\ m=1,\dots,M \quad (7)$$
$$y_{tr}\in\{0,1\} \quad t=1,\dots,T;\ r=1,\dots,R \quad (8)$$
$$v_{tmr}\in Z_+ \quad t=1,\cdots,T;\ m=1,\dots,M;\ r=1,\dots,R \quad (9)$$
$$S_{tim}\in Z_+ \quad t=1,\cdots,T;\ i=1,\dots,N; m=1,\dots,M \quad (10)$$
$$I_{tim}\in Z_+ \quad t=1,\cdots,T\ \ i=1,\dots,N,\ m=1,\dots,M \quad (11)$$
$$L_{tim}\in Z_+ \quad t=0,\cdots,T\ \ i=1,\dots,N,\ m=1,\dots,M. \quad (12)$$

The first part of the objective function (1) represents the profit from product sales, the second part represents the total fixed cost incurred due to deliveries on all routes across days of the planning horizon, the third part represents the costs of supplying products, the fourth part represents the total holding cost of inventory at the stores throughout the planning horizon, and the fifth component is the cost incurred due to sales lost.

Constraint (2) states that the total product deliveries to store m on day t cannot exceed the sum of the capacities allocated to store m from all delivery routes serving store m on that day. Constraint (3) indicates that if a delivery on day t is made from route r, the capacity of the truck B will be allocated to stores covered by route r. If the shipment is not made, no delivery will occur to the stores from this route. Constraint (4) specifies that the initial inventory at stores is zero. Constraint (5) calculates the end-of-day inventory by subtracting the number of products sold from the previous day's inventory plus the received products. Constraint (6) ensures that the sales quantity for each product and store does not exceed daily demand, as demand is shared by two non-negative integers according to this constraint. Constraints (5) and (6) together indicate that sales cannot exceed the minimum of inventory at hand including daily replenishments and the daily demand, and when the former is smaller, number of sales lost complements the difference. (7), (10)-(12) ensure that daily delivery quantities of products to stores, sales/lost sales amounts, and inventory levels at stores are non-negative integers. The allocation of daily delivery capacities on routes to stores are also non-negative integers by (9). (8) specifies that the decision to make a delivery on a route on a certain day is a binary decision.

4. Sales Data Series Analysis with ARIMA and Artificial Neural Networks

For prediction, we employ both traditional statistical time series models and neural network-based models to forecast product sales across multiple stores. The traditional statistical approach involves ARIMA models. For predicting the sales process of each store product combination, an ARIMA model minimizing the Akaike Information Criterion (AIC) is selected. For sales prediction with neural networks, numerous architectures involving recurrent, convolutional, dense and other specific layers are devised. The goal is to identify the most effective neural network model for each store-product combination by comparing the Mean Squared Error (MSE) of predictions by each model. MSE is a more practical criterion on test data prediction error, which is suitable since the complexity of neural networks involves many distinct and layer type specific parameters in many layers. AIC is suitable for ARIMA models, as these models are more structural in their complexity being defined by degrees of autoregressive, integrated and moving average terms. AIC balances prediction quality with penalties on overly complex models.

The two approaches of prediction, ARIMA and neural networks are compared in this study, initially with respect to prediction performance in Subsection 4.3. These two approaches are used for predicting sales (d_{tim}) parameters of the integer program (1)-(12). The inventory replenishment decisions from solutions of (1)-(12) help explore the more broad and practical effects of prediction models on firm profitability, as discussed in Section 5.

4.1 ARIMA Model Selection

ARIMA models are a cornerstone in time series forecasting, renowned for their effectiveness in capturing the underlying patterns in both stationary and non-stationary data. ARIMA processes combine autoregressive terms, moving averages, and differencing to model complex time series behaviors such as consumption, pricing, investment [24] and demands [25] of commodities.

The optimal ARIMA model, typically denoted with parameters (p,d,q), where p represents the number of autoregressive terms, q the number of moving average terms, and d the degree of differencing, is selected based on the minimum AIC. This criterion helps balance model complexity against goodness-of-fit, aiming to minimize the AIC value to enhance predictive accuracy [26].

We utilize the Auto ARIMA function from the pmdarima package (version 2.0.4) in Python 3.10 to determine parameters, including the appropriate level of differencing. This function identifies the need for differencing by conducting stationarity tests such as the augmented Dickey-Fuller test as part of its process, adjusting the data as needed before model estimation [27].

External factors, such as weekends and holidays, significantly influence sales patterns. We incorporate these as exogenous regressors in our ARIMA models, aligning them with the forecast day to reflect the known status of these variables accurately. This approach acknowledges the dynamic influence of special events, which can be modeled separately as intervention events [28]. ARIMA models are robustly trained on historical sales data to forecast future demands, taking into account both typical sales patterns and special events that may disrupt these patterns. This comprehensive approach allows for more accurate and reliable forecasting, crucial for effective demand planning and inventory management. With the addition of the exogenous variable, the model is termed formally as ARIMAX (X-exogenous), however, in this study, we prefer referring as ARIMA to prediction models from this class.

Table 1 showcases the optimal ARIMA models for each store-product combination, determined by the Auto ARIMA function to minimize the AIC.

Table 1. *AIC minimizing ARIMA model parameters* (p, d, q) *for forecasting sales in analyzed store and product combinations.*

	Product 1	**Product 2**	**Product 3**	**Product 4**
Store 1	(1,1,1)	(1,0,1)	(0,1,5)	(1,0,3)
Store 2	(2,1,4)	(2,1,3)	(1,1,2)	(1,1,1)
Store 3	(2,1,3)	(1,0,1)	(1,0,1)	(0,1,1)

Common among the models is the necessity for differencing the series once to address non-stationarity. While the models generally feature low degrees of autoregressive terms, variations are evident in the moving average components—particularly for Product 3 in Store 1, which requires higher moving average terms. Similarly, the models for Product 1 in Stores 2 and 3 indicate a higher overall number of terms, aligning with the lowest AIC and suggesting a more complex model structure to capture the underlying patterns in the sales data.

The ARIMA model, long recognized for its effectiveness in modeling non-stationary time series, leverages past observations and random errors to predict future values, making it a robust tool in various real-world applications. However, the advent of increased computational power has made more complex deep learning architectures like Long Short-Term Memory (LSTM) increasingly feasible for handling time series data. These advanced models generally surpass traditional machine learning and statistical approaches like ARIMA in performance, thanks to their ability to capture and analyze more complex patterns and dependencies within the data [29].

4.2 Neural Network Architectures

In addition to the ARIMA models, we explored a range of neural network architectures tailored for time series forecasting. These architectures aim to capture temporal dependencies in the data, particularly complex patterns that appear in longer span of lags. Several neural network architectures are considered here, and tested on the sales data for prediction performance.

In the domain of time series forecasting, various neural network architectures have been devised to address the intricate patterns inherent in data sequences. These architectures leverage unique strengths of various layers to enhance prediction accuracy. LSTMs are pivotal for capturing chronological sequences, while Convolutional Neural Networks (CNNs) excel at identifying spatial dependencies by analyzing patterns across different features within the data. The integration of Bidirectional LSTMs allows the models to assimilate information from both past and future contexts, thereby enriching the understanding of data sequences [30]. Multi-Scale CNNs employ levels of filters to capture a broader spectrum of patterns, from minute details to overarching trends [31]. The introduction of dilated convolutions extends the receptive field while allowing for more efficient concurrent processing and encompassing nonlinear sequential patterns [32]. Inception modules within LSTM frameworks reduce computational complexities of convolutional filters, efficiently diversifying the feature detection before temporal analysis [33]. Additionally, attention mechanisms direct the model's focus to the most pertinent parts of the data, enhancing the relevancy of predictions [34], although this enhancement does not guarantee better RNN performance compared to CNNs [35]. Dropout layers interspersed within these networks help in preventing overfitting by randomly omitting subsets of features during training phases. By hybridizing these components—connecting and layering them in configurations—a higher potential for dissecting and predicting complex temporal patterns is aimed.

Table 2 summarizes neural network architectures devised for sales prediction. The table outlines each model's specific layers, configurations, and the role of each component in the overall architecture.

The comparison between ARIMA and neural network models allows for the identification of the most effective model for each store and product combination, based on the MSE on the test data.

Table 2. *Overview of Neural Network Architectures for Sales Demand Forecasting.*

Architecture	Description
*Conv1D(64, 3) → Conv1D(128, 3) → LSTM(128) → Dense(1)	**CNN + LSTM**: Uses convolutional layers to capture local patterns, followed by an LSTM layer to capture temporal dependencies.
Bidirectional(LSTM(128)) → Bidirectional(LSTM(128)) → Dense(1)	**Bidirectional LSTM**: Processes sequences in both forward and backward directions to capture context from both sides.
Conv1D(256, 1) → Conv1D(128, 3) → Conv1D(64, 5) → Concatenate() → Flatten() → Dense(1)	**Multi-Scale CNN**: Applies convolutional filters of different sizes to capture patterns at multiple scales, then concatenates outputs of convolutional layers, flattens and feeds into the final dense layer.
Conv1D(64, 3) → Dropout(0.25) → Conv1D(128, 3) → LSTM(128) → Dropout(0.25) → Dense(1)	**Enhanced CNN + LSTM**: Similar to the CNN + LSTM model, but with dropout layers added for regularization, improving the model's generalization capabilities.
Bidirectional(LSTM(128)) → Dropout(0.25) → Bidirectional(LSTM(128)) → Dense(128) → Dropout(0.25) → Dense(64) → Dropout(0.25) → Dense(1)	**Enhanced Bidirectional LSTM**: Expands on the Bidirectional LSTM by incorporating dropout layers and additional dense layers for enhanced feature extraction and prediction.
Conv1D(256, 1) + Conv1D(128, 3) + Conv1D(64, 5) → Concatenate() → Flatten() → Dropout(0.25) → Dense(128) → Dense(64) → Dropout(0.25) → Dense(1)	**Enhanced Multi-Scale CNN**: A multi-scale CNN with dropout and additional dense layers to better capture and process the input data.
Conv1D(128, 1) → Conv1D(64, 3) → Conv1D(128, 3) → Add() → LSTM(128) → Dropout(0.25) → Dense(64) → Dropout(0.25) → Dense(1)	**CNN-LSTM with Residual Connections**: Integrates residual connections to allow the model to learn more effectively, followed by LSTM and dense layers.
InceptionModule(64) → LSTM(128) → LSTM(128) → Dropout(0.25) → Dense(64) → Dropout(0.25) → Dense(1)	**Inception-LSTM Hybrid**: Combines an Inception-like module for capturing diverse patterns, with LSTM layers to model temporal sequences, followed by dropout and dense layers.
Conv1D(64, 3, dilation_rate=1) → Conv1D(128, 3, dilation_rate=2) → LSTM(128) → Dropout(0.25) → Dense(64) → Dropout(0.25) → Dense(1)	**Dilated CNN + LSTM**: Uses dilated convolutions to expand the receptive field, followed by LSTM and dense layers, allowing the model to capture broader temporal patterns.
Conv1D(64, 3) → Bidirectional(GRU(128)) → Attention → Flatten() → Dense(128) → Dropout(0.25) → Dense(64) → Dropout(0.25) → Dense(1)	**CNN-BiGRU with Attention**: Combines CNN and bidirectional Gated Recurrent Unit (GRU) with an attention mechanism to focus on relevant parts of the input, followed by dense layers for prediction.

*Arrows indicate order of layers, plus symbol indicates layers whose outputs are combined by a concatenation layer. Conv1D(filters, kernel size): 1D Convolutional Layer, where filters is the number of output filters, and kernel size is the length of the 1D convolution window. LSTM/GRU(units): LSTM/GRU layer with the parameter indicating number of hidden units. Bidirectional(LSTM/GRU(units)): A Bidirectional LSTM/GRU layer that processes input sequences in both forward and backward directions. Flatten(): Converts a multidimensional tensor into a single dimension. Dropout(rate): Regularization technique where a fraction rate of input units is set to 0 at each update during training. Concatenate(): Merges multiple layers into a single layer. Dense(units, activation): Fully connected layer where units is the number of neurons and activation is the activation function. InceptionModule(filters): A custom module inspired by the Inception architecture, used to capture multiple types of patterns in the input data. Dilated Convolutions: Convolutions with holes (dilations) to increase the receptive field without losing resolution. Attention Mechanism: Mechanism that allows the model to focus on specific parts of the input sequence when making predictions. All models end with a fully connected layer -Dense(1)- for prediction of the number of sales in the following day.

Both the ARIMA and neural network models were evaluated based on their ability to predict future sales at various horizons, specifically at +1, +5, +10, +15, +20, +25, and +30 days ahead. Multi-step ahead forecasting is necessary in this context because the predictions are to be used as inputs for a multi-period model designed for planning inventory replenishment. Accurate forecasts for these future periods are critical for informed decisions about stock replenishment, distribution, and managing inventory levels.

For neural network models, multi-step ahead forecasting is performed by generating predictions iteratively over a 30-day horizon. The predictions for each day were fed back into the model to predict the subsequent days, alongside the known values of the special day vector. For the ARIMA model, a similar rolling horizon approach was used. The model was updated iteratively, allowing for adaptive predictions over the 30-day horizon.

The observed trends in MSE across products with varying sales frequencies and volumes provide critical insights into the challenges of predictive modeling for different types of sales data. Notably, the variation in MSE trends between products with sporadic, almost binary sales patterns and those with consistent, higher-volume sales underscores the strengths and limitations of forecasting models under different conditions and prediction horizons. As an example generalizable to different neural network architectures and store-product combinations, consider the CNN+LSTM architecture predicting Product 2 at Store 1, which typically records low sales—often none or just one item per day. This product exhibits long periods without sales, punctuated by intervals of daily sales. In such scenarios, MSE remains relatively low, especially when compared to products that experience a broader range of daily sales volumes, such as Products 1, 3, and 4. Product 3, in particular, demonstrates extended periods without sales followed by bursts of high-volume sales days. These bursts make sales from one day somewhat predictive of the next, although transitions from sales to no-sales periods significantly impact the accuracy of predictions, causing MSE to increase as the prediction horizon extends (Fig. 2). In contrast, Products 1 and 4 experience a wide range of sales volumes but without the distinct no-sales/sales phases observed in Products 2 and 3. Additionally, markers for special days consistently boost sales, enhancing predictability. Surprisingly, this results in a decrease in MSE as predictions extend further into the future (Fig. 2), illustrating how consistent sales patterns, even at higher volumes, can enhance forecasting accuracy.

4.3 Model Evaluation and Comparison

Data was collected over a period of 640 days, from 27.12.2021 to 27.09.2023, across three stores and four product categories. The final 30 days of this period are reserved exclusively for computational testing of the optimization model's performance and are not included in the training or test datasets for the time series prediction models. Of the remaining 610 days, the first 550 days are utilized as the training data set, while the last 30 of subsequent 60 days serve as the validation data set. The initial 30 days of prediction model testing data is reserved as lag sequences for sequence-based prediction models involving layers such as LSTM or GRU.

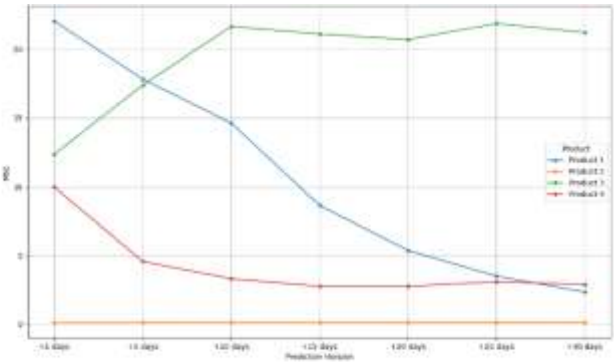

Figure 2. *MSE values for CNN+LSTM neural network model for +1, +5, +10, …, +30 day horizon predictions of Store 1; Products 1-4.*

The performance of each model is evaluated using MSE. This metric is calculated for each of the specified horizons (i.e., +1, +5, +10, +15, +20, +25, +30 days ahead). When evaluating the performance of neural network models across various prediction intervals, the model that consistently yields the lowest MSE is chosen. This method ensures that our forecasts extending through the planning horizon are based on the most accurate models available for each combination of store and product type. For each store and product type, we reviewed the MSE values across all available prediction horizons (e.g., +1, +5, +10, +15, +20, +25, +30 days ahead). The neural network model that recorded the lowest MSE for most of these prediction intervals was then selected as the optimal model for making forecasts. This strategy prioritizes overall performance across time, rather than exceptional performance at a single prediction interval. Table 3 presents the neural network architectures chosen to predict future sales for different store and product combinations. The Inception-LSTM Hybrid and CNN-BiGRU with Attention models do not appear as best model for any

Table 3. *Artificial Neural Network architectures selected for forecasting sales in analyzed store and products.*

	Product 1	Product 2	Product 3	Product 4
Store 1	Enhanced CNN + LSTM	Bidirectional LSTM	CNN-LSTM with Residual Connections	CNN + LSTM
Store 2	Dilated CNN + LSTM	Bidirectional LSTM	Dilated CNN + LSTM	CNN + LSTM
Store 3	Bidirectional LSTM	CNN + LSTM	Bidirectional LSTM	Enhanced Multi-Scale CNN

store-product pair. These sophisticated architectures integrate complex structures like inception modules and attention mechanisms in order to model datasets with intricate patterns and substantial variability. However, when tested against the dataset comprising 550 data points across two time series—sales and special days—these models are not selected due to their complexity potentially being disproportionate to the dataset size and complexity. The data, limited in scale and diversity, likely does not exhibit the high dimensional variability and the patterns that require extensive feature extraction capabilities that these models are designed to handle. For the dataset at hand, simpler RNN models, including their bidirectional versions and CNN-enhanced variants, proved sufficient. These models effectively capture the temporal dynamics without the extensive data requirements, computational overhead and potential overfitting associated with more elaborate architectures. This scenario underscores the importance of matching model complexity to data characteristics, where simpler models often yield robust performance without the complications introduced by more complex architectures.

The comparative analysis between ARIMA and selected neural network models across various store and product combinations reveals distinct trends in prediction accuracy, especially between next-day forecasts and those over longer horizons. For Store 1 and Store 2, ARIMA models show marginally better performance (i.e., lower MSE) than neural networks for next-day forecasts for two of the four products. This suggests that ARIMA's traditional time-series approach is slightly more effective for products with stable, predictable patterns in capturing short-term dynamics.

However, as the prediction horizon extends to +5, +10 days, and beyond, neural networks consistently outperform ARIMA models, with this advantage becoming particularly significant for Products 3 and 4 (Fig. 3). Neural networks excel in modeling complex nonlinear relationships and interactions [36], a capability that grows in importance with longer forecast periods. They effectively handle the variability and trends that ARIMA models often fail to capture, especially in datasets with volatile or irregular sales patterns.

This robustness of neural networks in diverse and challenging forecasting scenarios is evident in their superior performance over extended periods, as shown in Fig. 3. The divergence in MSE trajectories between the models in this figure illustrates neural networks' enhanced predictive capabilities for mid to long-range forecasting tasks. Notably, the figure excludes Product 3 for Store 1, as the ARIMA model's MSEs are disproportionately high compared to the more consistent scale of neural network MSEs for all products. Despite the high variability in Product 3's sales, CNN/LSTM architectures successfully predict sudden sales spikes, a task where the ARIMA model struggles due to reliance on historical error correlations a low number of lags that do not adequately anticipate such atypical sales spikes despite the high number of terms in the model.

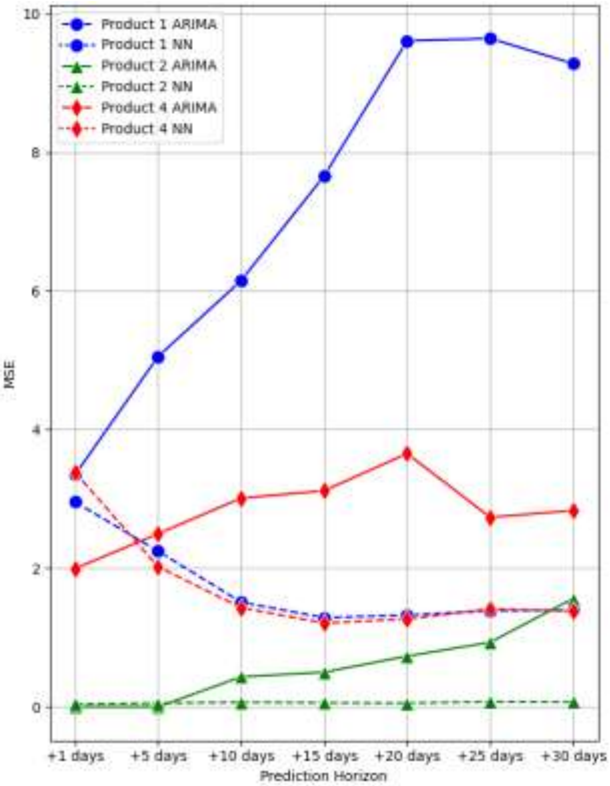

Figure 3. *Comparison of MSEs for the selected ARIMA and neural networks for predicting Product 1,...,4 sales at Store 1.*

The pattern in Fig. 3 generalizes as follows. When averaged over 12 store-product combinations, MSE for the neural network with minimum MSE has 6% lower MSE than the best model provided by Auto ARIMA. This gap grows to 60% when predicting 15 days ahead, and 74% when predicting 30 days ahead.

In conclusion, the selected neural network models demonstrate a superior ability to accurately predict sales for planning horizons ranging from 5 to 30 days. This capability makes them particularly suitable for informing our optimization model, which is designed to make efficient inventory decisions within this timeframe.

In the following section, we will explore how the performance of these forecasting models influences firm profitability through replenishment decisions made using the optimization model.

5. Computational Results

The Integer Programming model for store inventory replenishment optimization is coded on Python 3.10 and solved with Gurobi 11, running on a computer equipped with an AMD Ryzen Threadripper 3960X 24-Core Processor and 64 GB RAM.

In our computational experiments, we utilize anonymized price, unit cost and sales data provided under a non-disclosure agreement with a confidential industry partner, ensuring that the economic parameters accurately reflect real-world business conditions. Additionally, route costs are modeled based on distance and time expenditures, aligning with the realities of transportation. Inventory holding costs and lost sales penalties are established based on assessments by the company's staff and industry experts, providing a realistic basis for operational simulations. Importantly, as a matter of firm policy, item prices and costs are fixed seasonally, which allows us to isolate the demand process and assess the impact of predictive accuracy on operational performance more clearly.

The model utilizes a dataset spanning 640 days, with the final 30 days reserved specifically for jointly evaluating the performance of the sales prediction models and the inventory replenishment optimization. This evaluation is critical for assessing the profitability of the decisions derived from predicted sales and the optimal replenishment and inventory strategies as defined by the integer programming model equations (1)-(12).

For each testing phase, a fixed planning horizon of T days is established. Sales predictions for these T days are generated using neural network (the prediction by selected neural network for the specific store-product combination is abbreviated as NN in this section) models (alternatively, ARIMA models), which have been trained on data up to the first day of the model testing period for each store and product combination. The predicted sales then serve as the demand input for the Integer Programming model (1)-(12). Once the optimization problem is solved, the replenishment orders for the next day are executed according to the model's outputs (from the optimal solution, decisions for $t=1$ are executed daily). The actual sales for the day are accounted for according to initial inventory (uniformly 0 for all store and products), replenishment decisions from (1)-(12) and actual sales quantities according to the real data reserved for testing.

On the subsequent day, the inventory status is updated based on the outcomes of the previous day's prediction-optimization activities and the actual sales data. This process repeats daily for 30 testing days: the prediction models are updated with new real sales data (both neural network and ARIMA are incrementally trained with each day's additional actual sales data as would be executed in an overnight run for daily decisions), and the optimization model (1)-(12) is rerun for the next T-day horizon with updated beginning inventory and new demand predictions. Thus, the cycle of daily sales and end-of-day inventory level computation continues, which depends on preceding day inventory status, optimal replenishment ($t=1$) decisions from (1)-(12) and actual sales data.

This methodical approach allows for continuous evaluation of the predictive accuracy and optimization efficacy over a period, ensuring that each day's operations are informed by the most recent data and adjusted decisions during the measurement.

For each planning horizon of $T \in \{5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 30\}$ days, and for each of 12 store-product combinations, our training process involves an initial training of 100 epochs with early stopping implemented, featuring a patience of 30 epochs and a minimum learning rate of $1e$-5. Subsequently, the model undergoes 29 updates, each followed by a re-evaluation using early stopping with a reduced patience of 10 epochs and the same minimum learning rate. After each training and update phase, predictions for the respective planning horizon are generated. These predictions then serve as inputs for solving the optimization model specified in equations (1)-(12). For ARIMA, each planning horizon involves selecting and training 12 ARIMA models by Auto ARIMA, followed by 29 updates of this selected model. Similarly, future predictions covering the planning horizon and optimization run for each test day is made after training/updates. The entire computational process for 360 prediction model trainings/updates (12 models, 30 days, 360 training/updates in total), and 30 solutions of (1)-(12) for the 13 planning horizons took 33,402 seconds for NN and 23,613 seconds for the ARIMA case.

The computational analysis compares the performance of NN and ARIMA models over various planning horizons, focusing on their impact on total profit, gross profit and cost aspects. Total profit is calculated by deducting from the revenue all cost components including sales loss and operational costs of logistics and inventory. Gross profit considers only supply costs of items, thus is the difference of total revenue and the total cost of supplying the items sold. This comparison is visually represented in Fig. 4 and 5, which depict the trends in total and gross profit, and logistics, inventory and lost sales costs across different planning windows for both models.

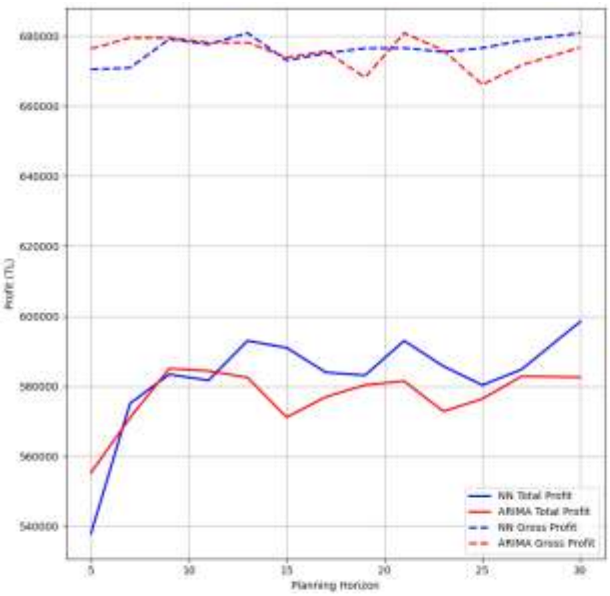

Figure 4. *Total profit and gross profit of the company over 30 days of applying optimal replenishment decisions based on NN and ARIMA predictions.*

NN exhibited significant improvements in total profit, especially at the longer planning windows of 13-30 days, achieving total profits ranging from 544,230 Turkish Liras (TL) to 609,650 TL. In shorter planning horizons (5-7 days), however, these models incur high logistics costs (Fig. 5) due to frequent replenishment orders (Fig. 6). This early inefficiency is mitigated by a 15-day planning horizon, where costs balance out with less frequent replenishments and higher inventory levels. Notably, neural networks are adept at anticipating sudden increases and long-term trends in order amounts, optimizing supply chain dynamics over longer horizons with higher sales levels (Fig. 4, gross profit) lower inventory and lost sales costs (Fig. 5).

Neural networks particularly demonstrate a capability to maintain sales levels without a reducing trend, indicating their effectiveness in capturing high future sales potential in long-range forecasts. In contrast, ARIMA models, despite predicting trends, exhibit fluctuations in their ability to match the sales levels required for optimal profitability in extended forecasts. ARIMA models are slower to adapt to sudden market changes, often resulting in delayed responses to sharp increases in sales. Predicted sales levels tend to underestimate actual demand in longer horizons, affecting overall profitability (Fig. 5). For all planning horizons, this adds up to 0.9% lower profits when using ARIMA models. When restricting to longer horizon planning (T>13), which avoids inefficient frequent replenishments for both prediction methods, prediction using neural networks elicits 1.6% higher profits on average, when optimal decisions from (1)-(12) is applied on historical sales data. The difference is approximately 2.3% when NN with T=30 is compared to ARIMA with T=9, where both attain their respective highest total profits.

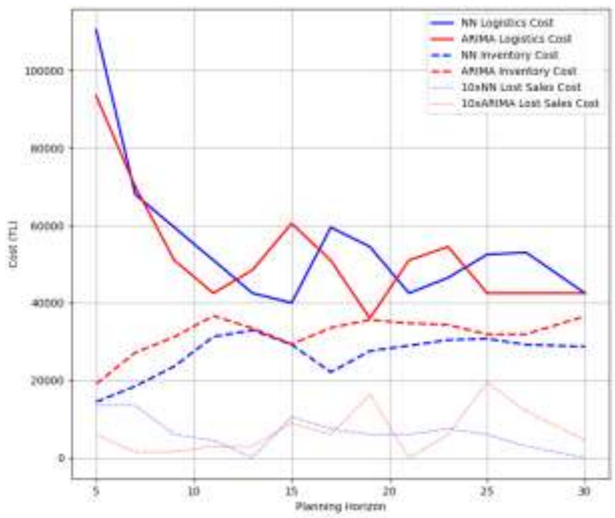

Figure 5. *Total logistics, inventory holding and lost sales costs over 30 days of optimal replenishment decisions based on NN and ARIMA predictions. Lost sales cost figures are scaled up tenfold for visibility.*

ARIMA models struggle relatively more with cost efficiency, less so for logistics costs, but particularly in managing lost sales and inventory in longer planning horizons.

Both models primarily utilize the route serving all three stores throughout the planning periods. In longer windows, strategies include occasional utilization of routes serving a single or two stores to rebalance inventory against prolonged demand periods. The anticipation of demand surges by NN and short planning horizons result in an increased frequency of 13 three-stores replenishments in 30 days (Fig. 6) reflecting as high delivery costs and reduced profit for NN with $T = 5$. In shorter horizons, frequent replenishment strategies result in higher logistics costs, particularly for NN. This pattern gradually improves as the planning horizon is extended, with both models showing better cost management with less frequent replenishments.

Replenishment frequencies from the store perspective align with the dispatching route perspective, as predominantly all stores are served in the same day (Fig. 7, bars). As expected, a reduction in delivery frequencies and an increase in inventory levels coincide with an increase in the number of products per delivery (Fig. 7, dashed line plots). Given that delivery methods incur fixed charges, optimizing the number of products per delivery emerges as a crucial component of cost-saving strategies in longer planning horizons, effectively reducing the per-item logistics cost.

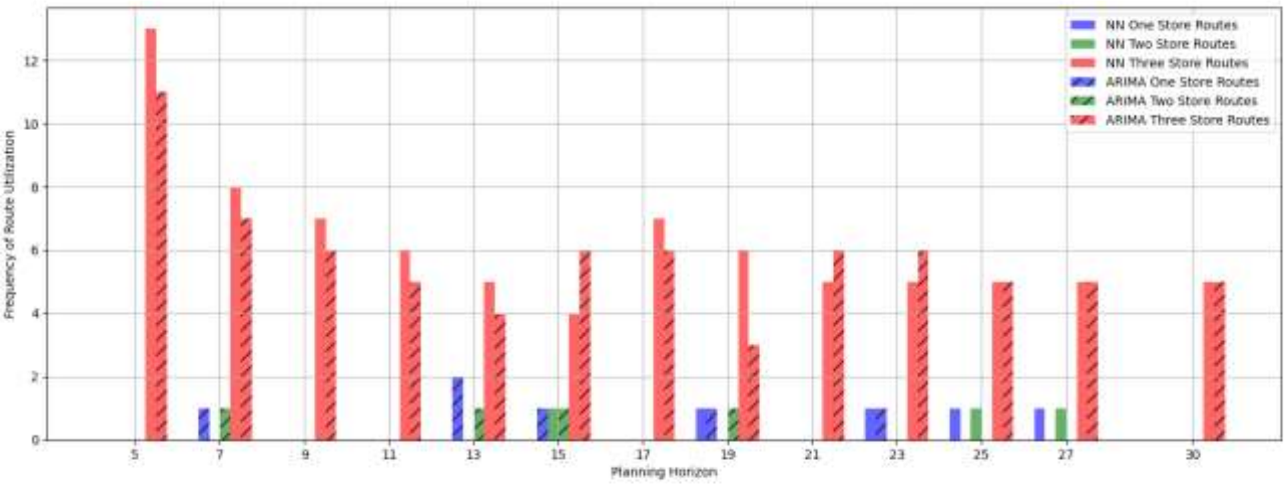

Figure 6. *Number of times routes serving to a single store, two and three stores is used throughout the 30 test days as per the optimal replenishment decisions based on NN and ARIMA predictions.*

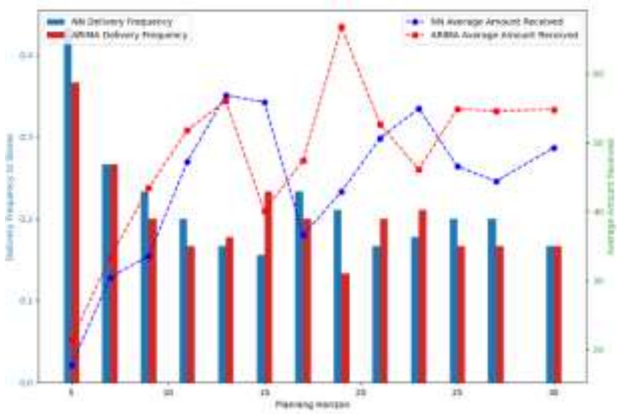

Figure 7. *Average frequency of delivery to stores in rate of deliveries per day (bars), and the average number of items delivered to a store per delivery (dashed lines).*

6. Conclusions

This study highlights the critical importance of selecting an appropriate prediction method that aligns with the characteristics of the data at hand. By comparing traditional statistical time series methods with advanced deep learning architectures, we have demonstrated the substantial benefits of integrating predictive models with optimization tools to enhance practical applicability in inventory management.

Better predictions improve the performance of optimized decisions. This is empirically validated by applying decisions derived from the prediction-optimization framework against historical real data, where the performance is assessed in the context of real-world operational flows. Such empirical assessments are vital for industrial applications that rely on the accuracy of these predictions to make informed decisions that directly impact the financial health and operational efficiency of organizations.

Moreover, the study delves into the integration of sophisticated forecasting methods with inventory management decisions, illustrating how enhancements in forecasting with neural network architectures can be leveraged to refine replenishment strategies that improve firm profitability. This integration is particularly valuable in demonstrating how optimized inventory management, driven by precise demand forecasts, contributes to cost savings and improved service levels.

The framework presented here, as summarized in Fig. 1, provides a foundation for predicting sales and optimizing inventory management that can be adapted to a variety of settings beyond the computational case considered. An application should start with a thorough determination of which items and locations require the collection of historical sales data. This is crucial as the demand dynamics might differ significantly across products and markets. Unlike the seasonally fixed supply costs and retail prices in the case presented, some settings may require the forecasting of future prices and other cost parameters dynamically. This is particularly relevant where costs and prices fluctuate due to market conditions, regulatory changes, or external economic factors. Parameter definitions in (1)-(12) are flexible and allow for such variability. Modifications to the integer programming model (1)-(12) may be necessary to accommodate specific characteristics of the inventory being managed. For instance, scenarios involving multiple warehouses or more complex configurations of facilities would require adjustments to index, variable, and constraint definitions. Similarly, dealing with perishable items would necessitate extensions to the model to account for varying shelf lives and different regulatory requirements; possibly involving stochastic parameters and chance constraints. With the necessary adaptations, the prediction-optimization framework remains effective across different sectors and retail models.

While this study considers univariate forecasting models, future research could explore the potential of multivariate modeling. This could include VARMA models on the statistical side and multivariate deep neural networks on the deep learning side. Additionally, there is a promising avenue for extending this with stochastic modeling. The readily available probabilistic error models from statistical time series can be paralleled with model-specific error modeling for neural network predictions. Such advancements could further enhance the robustness and reliability of forecasting models, thereby supporting more dynamic and responsive inventory management strategies.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have influenced this study.
- **Acknowledgement:** We are grateful to the industry partner for providing price, unit cost and sales data, anonymized for confidentiality, but crucial in ensuring that our study accurately reflects real-world retail conditions. We also extend our sincere thanks to the referees whose insightful comments and suggestions significantly enhanced the quality and clarity of this manuscript. Their meticulous critiques and constructive feedback were invaluable in refining our analysis and interpretations
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to confidentiality.

References

[1] Mukherjee, P., & Bose, S. (2008). Does the Stock Market in India Move with Asia?: A Multivariate Cointegration-Vector Autoregression Approach. *Emerging Markets Finance and Trade*, *44*(5), 5–22. https://doi.org/10.2753/REE1540-496X440501.

[2] Michalski, G. M. (2013). Value-Based Inventory Management. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1081276.

[3] Muckstadt, J. A., & Sapra, A. (2010). *Principles of inventory management: when you are down to four, order more*. Springer.

[4] Wild, A. (2018). *Best practice in inventory management* (3 Edition). Routledge.

[5] Silver, E. A., Pyke, D. F., Peterson, R., & Silver, E. A. (1998). *Inventory management and production planning and scheduling* (3. ed). Wiley.

[6] Chandramohan, J., Asoka Chakravarthi, R. P., & Ramasamy, U. (2023). A comprehensive inventory management system for non-instantaneous deteriorating items in supplier- retailer-customer supply chains. *Supply Chain Analytics*, *3*, 100015. https://doi.org/10.1016/j.sca.2023.100015.

[7] Yang, H.-L. (2023). An optimal replenishment cycle and order quantity inventory model for deteriorating items with fluctuating demand. *Supply Chain Analytics*, *3*, 100021. https://doi.org/10.1016/j.sca.2023.100021.

[8] Rinaldi, M., Fera, M., Macchiaroli, R., & Bottani, E. (2023). A new procedure for spare parts inventory management in ETO production: a case study. *Procedia Computer Science*, *217*, 376–385. https://doi.org/10.1016/j.procs.2022.12.233.

[9] Gutiérrez, J., Colebrook, M., Abdul-Jalbar, B., & Sicilia, J. (2013). Effective replenishment policies for

the multi-item dynamic lot-sizing problem with storage capacities. *Computers & Operations Research*, *40*(12), Article 12. https://doi.org/10.1016/j.cor.2013.06.007.

[10] Yang, H.-L. (2012). Two-warehouse partial backlogging inventory models with three-parameter Weibull distribution deterioration under inflation. *International Journal of Production Economics*, *138*(1), 107–116. https://doi.org/10.1016/j.ijpe.2012.03.007.

[11] Chiu, S. W., Wu, C.-S., & Tseng, C.-T. (2019). Incorporating an expedited rate, rework, and a multi-shipment policy into a multi-item stock refilling system. *Operations Research Perspectives*, *6*, 100115. https://doi.org/10.1016/j.orp.2019.100115.

[12] Kumar, S., & Mahapatra, R. P. (2021). Design of multi-warehouse inventory model for an optimal replenishment policy using a Rain Optimization Algorithm. *Knowledge-Based Systems*, *231*, 107406. https://doi.org/10.1016/j.knosys.2021.107406.

[13] Roozbeh Nia, A., Hemmati Far, M., & Akhavan Niaki, S. T. (2014). A fuzzy vendor managed inventory of multi-item economic order quantity model under shortage: An ant colony optimization algorithm. *International Journal of Production Economics*, *155*, 259–271. https://doi.org/10.1016/j.ijpe.2013.07.017.

[14] Mareeswaran, M., & Anandhi, M. (2021). Optimization of inventory in agriculture material processing industry by using multi-item deterministic model. *Materials Today: Proceedings*, *46*, 4183–4186. https://doi.org/10.1016/j.matpr.2021.02.747.

[15] Nobil, A. H., Nobil, E., Afshar Sedigh, A. H., Cárdenas-Barrón, L. E., Garza-Núñez, D., Treviño-Garza, G., Céspedes-Mota, A., Loera-Hernández, I. de J., & Smith, N. R. (2024). Economic production quantity models for an imperfect manufacturing system with strict inspection. *Ain Shams Engineering Journal*, *15*(5), 102714. https://doi.org/10.1016/j.asej.2024.102714.

[16] Alon, I., Qi, M., & Sadowski, R. J. (2001). Forecasting aggregate retail sales:: a comparison of artificial neural networks and traditional methods. *Journal of Retailing and Consumer Services*, *8*(3), 147–156. https://doi.org/10.1016/S0969-6989(00)00011-4.

[17] Caglayan, N., Satoglu, S. I., & Kapukaya, E. N. (2020). Sales Forecasting by Artificial Neural Networks for the Apparel Retail Chain Stores. In C. Kahraman, S. Cebi, S. Cevik Onar, B. Oztaysi, A. C. Tolga, & I. U. Sari (Eds.), *Intelligent and Fuzzy Techniques in Big Data Analytics and Decision Making* (pp. 451–456). Springer International Publishing. https://doi.org/10.1007/978-3-030-23756-1_56.

[18] Das, P., & Chaudhury, S. (2007). Prediction of retail sales of footwear using feedforward and recurrent neural networks. *Neural Computing and Applications*, *16*(4), 491–502. https://doi.org/10.1007/s00521-006-0077-3.

[19] Penpece, D., & Elma, O. E. (2014). Predicting Sales Revenue by Using Artificial Neural Network in Grocery Retailing Industry: A Case Study in Turkey. *International Journal of Trade, Economics and Finance*, *5*(5), 435–440. https://doi.org/10.7763/IJTEF.2014.V5.411.

[20] Loureiro, A. L. D., Miguéis, V. L., & da Silva, L. F. M. (2018). Exploring the use of deep neural networks for sales forecasting in fashion retail. *Decision Support Systems*, *114*, 81–93. https://doi.org/10.1016/j.dss.2018.08.010.

[21] Deraz, N. (2023). Economic Order Quantity Predictive Model Using Supervised Machine Learning for Inventory Management of Fast-Moving Consumer Goods Distributors. *Plymouth Business School Theses*. https://doi.org/10.24382/2668.

[22] Kilimci, Z. H., Akyuz, A. O., Uysal, M., Akyokus, S., Uysal, M. O., Atak Bulbul, B., & Ekmis, M. A. (2019). An Improved Demand Forecasting Model Using Deep Learning Approach and Proposed Decision Integration Strategy for Supply Chain. *Complexity*, *2019*(1), 9067367. https://doi.org/10.1155/2019/9067367.

[23] Borade, A. B., & Bansod, S. V. (2011). Neural networks based vendor-managed forecasting: a case study. *International Journal of Integrated Supply Management*. https://www.inderscienceonline.com/doi/10.1504/IJISM.2011.040713.

[24] Jiang, S., Yang, C., Guo, J., & Ding, Z. (2018). ARIMA forecasting of China's coal consumption, price and investment by 2030. *Energy Sources, Part B: Economics, Planning, and Policy*, *13*(3), Article 3. https://doi.org/10.1080/15567249.2017.1423413.

[25] Dey, B., Roy, B., Datta, S., & Ustun, T. S. (2023). Forecasting ethanol demand in India to meet future blending targets: A comparison of ARIMA and various regression models. *Energy Reports*, *9*, 411–418. https://doi.org/10.1016/j.egyr.2022.11.038.

[26] Chyon, F. A., Suman, M. N. H., Fahim, M. R. I., & Ahmmed, M. S. (2022). Time series analysis and predicting COVID-19 affected patients by ARIMA model using machine learning. *Journal of Virological Methods*, *301*, 114433. https://doi.org/10.1016/j.jviromet.2021.114433.

[27] Ediger, V. Ş., & Akar, S. (2007). ARIMA forecasting of primary energy demand by fuel in Turkey. *Energy Policy*, *35*(3), Article 3. https://doi.org/10.1016/j.enpol.2006.05.009.

[28] Ďurka, P., & Pastoreková, S. (2012). ARIMA vs. ARIMAX–which approach is better to analyze and forecast macroeconomic time series. *Proceedings of 30th International Conference Mathematical Methods in Economics*, *2*, 136–140.

[29] Siami Namini, S., & Siami Namin, A. (2018). *Forecasting Economics and Financial Time Series: ARIMA vs. LSTM*.

[30] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681. IEEE Transactions on Signal Processing. https://doi.org/10.1109/78.650093.

[31] Deng, Z., Wang, B., Xu, Y., Xu, T., Liu, C., & Zhu, Z. (2019). Multi-Scale Convolutional Neural Network With Time-Cognition for Multi-Step Short-Term Load Forecasting. *IEEE Access*, *7*, 88058–88071.

IEEE Access. https://doi.org/10.1109/ACCESS.2019.2926137.

[32] Li, Y., Li, K., Chen, C., Zhou, X., Zeng, Z., & Li, K. (2021). Modeling Temporal Patterns with Dilated Convolutions for Time-Series Forecasting. *ACM Trans. Knowl. Discov. Data*, *16*(1), 14:1-14:22. https://doi.org/10.1145/3453724.

[33] Wang, J., Wang, W., Wei, S., Zeng, Y., & Luo, F. (2019). Time Series Sequences Classification with Inception and LSTM Module. *2019 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA)*, 51–55. https://doi.org/10.1109/ICTA48799.2019.9012862.

[34] Shih, S.-Y., Sun, F.-K., & Lee, H. (2019). Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, *108*(8), 1421–1441. https://doi.org/10.1007/s10994-019-05815-0.

[35] Zhou, K., Wang, W., Hu, T., & Deng, K. (2020). Time Series Forecasting and Classification Models Based on Recurrent with Attention Mechanism and Generative Adversarial Networks. *Sensors*, *20*(24), Article 24. https://doi.org/10.3390/s20247211.

[36] Durdu, D. (2010). A hybrid neural network and ARIMA model for water quality time series prediction. *Engineering Applications of Artificial Intelligence*, *23*(4), Article 4. https://doi.org/10.1016/j.engappai.2009.09.015.