Causal Inference-Driven Time-Series Forecasting for Inventory Optimization: A Time-Series Mixer Approach

Nguyen Thi Ngoc Anh[a,b,*], Nguyen The Phong[a,b], Nguyen Thi Ha[a,b]

[a]*Faculty of Mathematics and Informatics, Hanoi University of Science and Technology (HUST), No. 1, Dai Co Viet Street, Bach Mai Ward, Hanoi, 100000, Vietnam*

[b]*Institute for Digital Technology and Economy (BKFintech), HUST*

Abstract

Cost optimization and understanding causal relationships in operations are essential components of intelligent supply chain management, playing a crucial role in production, procurement, inventory control, and logistics. Accurate forecasting and causal explanations help optimize resource allocation and ensure effective responses to fluctuating customer demands. By balancing accuracy, adaptability and causal inference, a proposed model seamlessly combined Granger approach, Welford Time-Series Mixer (WTSMixer) forecasting and Reorder Point, Order Quantity Policy. To enhance forecasting accuracy under uncertain conditions, Time-Series Mixer provides a robust forecasting framework that optimizes supply chain operations in dynamic environments. Additionally, the Reorder Point, Order Quantity Policy replaces the traditional Economic Order Quantity method with reorder points and safety stock calculations based on demand forecasts, reducing stockouts and improving cost efficiency. Experimental results confirm the effectiveness of this approach, with the Welford Time-Series Mixer model demonstrating cost savings, validated through testing and real-world data comparisons.

Keywords: Forecasting, Inventory, Supply chain management, Uncertainty modelling, Causal inference

1. Introduction

Amid accelerating digitalization, the rise of e-commerce has reshaped retail, expanding product assortments and increasing the number of stock-keeping units (Wang et al., 2024). Modern supply chains are inherently complex and uncertain due to factors such as partner operations, customer behavior, competitor strategies, emerging technologies, and new product development (Abolghasemi et al., 2020). These uncertainties contribute to demand volatility, which poses a major challenge for supply chain managers, potentially leading to forecasting errors, upstream disruptions, and unnecessary costs (Abolghasemi et al.,

*Corresponding author: anh.nguyenthingoc@hust.edu.vn (Nguyen Thi Ngoc Anh)

Email addresses: anh.nguyenthingoc@hust.edu.vn (Nguyen Thi Ngoc Anh), thephong.hust@gmail.com (Nguyen The Phong), ha.nt242564m@sis.hust.edu.vn (Nguyen Thi Ha)

2020). To mitigate these negative consequences, it is essential to integrate advanced forecasting algorithms capable of predicting demand at a granular product level over short time horizons (Sohrabpour et al., 2021; Wang et al., 2024). A new area of growing interest in macroeconomic and financial forecasting is the combination of predictive densities (Virbickaitė et al., 2025).

Forecasting is crucial for decision-making across various business functions, including marketing, sales, production, procurement, finance, and accounting (Sohrabpour et al., 2021), (Coroneo and Iacone, 2025). Additionally, it serves as a foundational element in supply chain planning and strategy development, impacting production and logistics at all operational levels (Fildes et al., 2022). Consequently, a key operational tool for enhancing inventory control and raising consumer satisfaction is supply prediction (Mitra et al., 2022).

Classical time-series methods (e.g., ARIMA, SARIMA) (Sengupta et al., 2025) are standard for univariate forecasting but struggle with complex real-world data that contain cross-variable dependencies (Wang et al., 2024). Modern ML/DL approaches such as LSTM (Farsi et al., 2021), TFT (Lim et al., 2021), N-BEATS (Oreshkin et al., 2021), and TimesNet (Souto, 2024) have delivered strong results for multivariate forecasting. Nevertheless, these models are prone to overfitting, particularly when the target variable exhibits weak correlations with other features. Recent studies have highlighted the robustness of linear models in capturing time-dependent relationships (Liu et al., 2024). Building on this, TSMixer extends these capabilities by integrating linear models with nonlinear components, creating a multi-layer neural network that leverages highly correlated variables to enhance predictive performance.

In real-world systems driven by causality, correlation alone is insufficient to establish causal relationships. Causal inference has gained significant attention across multiple research domains, spanning philosophy, economics, biology, and data science (Guo et al., 2024). A central difficulty is disentangling causal structure—deciding whether effects are one-way or reciprocal, and separating them from associations induced by unmeasured confounders (Benkő et al., 2024).

Applying causal inference techniques in time-series forecasting enables researchers to filter out irrelevant variables, thereby improving prediction accuracy. Several studies have explored causal inference methodologies, including Variable Sensitivity Analysis (Sohrabpour et al., 2021), the Causal Interventional Prediction System (Chu et al., 2024), Granger causality (Granger, 1969), and Bayesian models (Gao et al., 2024). These techniques boost predictive performance by including only drivers causally linked to the target.

The following section summarizes the principal contributions of this work:

- Causal inference is employed through Granger causality testing to identify truly causal features for demand forecasting. This enables the construction of three scenario-based feature sets (Granger-Level 1, Level 2, Level 3), which demonstrate that selecting first-order causal variables consistently yields the most cost-effective outcomes.
- A novel WTSMixer architecture is proposed, integrating Welford normalization into the original

TSMixer framework to stabilize training, reduce variance, and improve generalization on small and volatile datasets.

- The forecasting model is optimized with respect to economic impact, using Total Inventory Cost as the loss function instead of traditional forecast error metrics. This approach aligns forecasting with downstream decision-making in inventory control.
- The entire framework is validated on real-world multivariate time series data, demonstrating that the integration of causal reasoning and cost-sensitive forecasting leads to significant improvements in operational efficiency over both traditional and deep learning benchmarks.

The structure of this work is as follows: In Section 1, introduce a causal inference-driven time series forecasting for inventory optimization. The relevant literature on causal inference, the forecasting of demand, inventory management optimization, and a suggested method in Section 3 are presented in Section 2. Experiments conducted on actual datasets are detailed in Section 4. Lastly, Section 5 summarizes the core outcomes and identifies avenues for further investigation.

2. Related Works

Causal inference, focused on identifying cause-and-effect relationships, is a central theme across these papers, as summarized in Table 1. An N-gram Causal Inference algorithm is introduced, offering a resilient n-gram–based causal estimator that performs matching across multiple models and generates pseudo-samples via Longest Common Subsequence, optimized using binary cross-entropy together with an n-gram loss (Guo et al., 2024). Techniques include time-varying Granger causality (Mrabet et al., 2025) and the Granger JKS method in both bivariate and multivariate forms (Raifu et al., 2025). In environmental science, methods like Peter and Clark Momentary Conditional Independence from Tigramite (Nitish and Indu, 2025), and Convergent Cross-Mapping based on Empirical Dynamic Modelling and Takens' theorem (Zhou et al., 2025), help uncover true causality in nonlinear systems. The Causal Feature Selection algorithm, grounded in Pearl's theory, identifies direct causes from observational data (Zhang et al., 2014). A Bayesian approach compares topologically embedded time series to infer causal types using Takens' theorem (Benkő et al., 2024). Lastly, a hybrid deep learning-causal framework uses Structural Causal Model Framework with deep neural networks for propensity scores and BNANs for causal effect estimation (Bodendorf et al., 2023). Building on these methods, causal inference offers a more robust understanding of underlying factor relationships beyond simple correlations. Its applications in forecasting through feature selection or integration with machine learning demonstrate strong potential to enhance accuracy and support decisions grounded in true causal effects.

Table 1: Overview of studies on causal inference.

Source	Year	Causal Inference Method	Research Objective	Problem Scope
(Zhang et al., 2014)	2014	Causal Feature Selection	Identify the features that have a direct impact on stock prices, helping the model make more accurate predictions	Annual financial statements of A-share firms listed on the Shanghai Stock Exchange
(Bodendorf et al., 2023)	2023	Structured Causal Inference	Analyzing causal relationships between interventions in supply chains while predicting supply chain disruptions	Supply chain disruptions
(Benkő et al., 2024)	2024	Bayesian Causal Inference	Distinguishing and assigning probabilities to causal relationships in dynamic systems	Epilepsy research using EEG data
(Guo et al., 2024)	2024	N-Gram-based Causal Inference	Applying causal inference to identify and explain the causes of accidents.	Accident analysis and prevention
(Mrabet et al., 2025)	2025	Granger Causality	Identify the dynamic causal relationship between economic variables	Data economic variables from the United States and the European Union
(Raifu et al., 2025)	2025	JKS Granger non-causality	Analyze the relationship between economic development and renewable energy	Panel dataset covering 35 OECD member countries
(Zhou et al., 2025)	2025	Convergent Cross Mapping	Causal relationship between air pollution factors, meteorological conditions, and the risk of ischemic stroke.	China High Air Pollutants daily dataset
(Nitish and Indu, 2025)	2025	Peter and Clark Momentary Conditional Independence	Causal relationship between LWST and water clarity is strongest in arid climate regions	GLWD and OpenStreetMap dataset

Forecasting in complex and dynamic environments such as supply chains, energy systems, or finance is often challenged by issues like non-stationarity, feedback loops, and latent confounders. Traditional forecasting methods typically rely on identifying statistical relationships such as correlations between variables (Zhao et al., 2024). While deep learning models are widely used to capture these dependencies (Sharma et al., 2024), they often result in black-box models where the learned relationships are opaque and lack interpretability (Poletaev et al., 2024). A fundamental limitation of these approaches is that correlation does not imply true causal influence (Zhao et al., 2024), which can lead to models that are less robust to noise, distribution shifts, and new environments (Poletaev et al., 2024). Moreover, feature selection meth-

ods based purely on correlation or direct predictive utility may overlook important underlying causal drivers (Tian et al., 2024).

To address these limitations, causal inference focuses on uncovering cause-effect relationships that are more stable and generalizable across different environments (Sharma et al., 2024). A prominent approach is Causal Feature Selection (Zhao et al., 2024). Techniques such as Granger Causality (Poletaev et al., 2024), Convergent Cross-Mapping (Sharma et al., 2024), and PCMCI (Zhang et al., 2024) are employed to discover causal links and identify features that have direct or indirect influence on the target variable. Utilizing causally-selected features has been shown to reduce input dimensionality, improve model interpretability, and enhance forecasting performance (Zhao et al., 2024), as demonstrated in applications such as wind power and water quality prediction. Beyond feature selection, causal principles can be further integrated into model architectures using mechanisms like sample weighting or simulated interventions to remove spurious correlations and improve robustness, particularly in situations with scarce historical data or distributional shifts (Yang et al., 2025). Additionally, further advancements in embedding causal knowledge into predictive frameworks to enhance model stability and performance have been discussed by (Fafoutellis and Vlahogianni, 2025).

In order to balance service quality and cost effectiveness, forecasting consumption and management of stocks must be integrated. (van der Haar et al., 2024). Traditionally, these tasks have been treated separately, often using a two-stage "forecast-then-optimize inventory " approach. However, improving forecast accuracy using standard metrics such as Mean Squared Error does not necessarily result in better inventory decisions, partly due to the asymmetry between the costs of overstocking and stockouts. This disconnect highlights the gap between forecasting objectives and inventory performance goals.

Recent methods aim to directly optimize inventory outcomes rather than solely focus on forecasting accuracy. One key direction is supervised learning for inventory control, where models are trained using cost functions that explicitly incorporate holding, shortage, or perishing costs. Various machine learning models have been explored, including MLPs, LSTMs, 1D-CNNs, and ensemble methods such as RNN-LSTM (Sukolkit et al., 2024). Gradient-based models like LightGBM have also been employed with inventory-specific cost functions that support efficient optimization (van der Haar et al., 2024). These approaches demonstrate that optimizing for inventory metrics can lead to superior performance in terms of total cost and service levels, even if raw forecast accuracy is not maximized (Seyedan et al., 2023). Furthermore, the use of Downstream Demand Inference together with basic time-series forecasting methods (Weighted Moving Average and Simple Moving Average) can enhance inventory performance in decentralized and multiechelon supply chains (Tliche et al., 2020).

An overview of studies combining causal inference, forecasting, and inventory is summarized in Table 2. Integrating causal inference, forecasting, and inventory optimization into a unified framework presents a promising direction for dynamic decision-making systems. Rather than relying solely on correlative signals, incorporating causally-relevant features into forecasting models leads to greater robustness, interpretability, and adaptability to changing environments. When these forecasts are directly connected to inventory

policies through cost-sensitive optimization objectives, the system transitions from reactive response to proactive control.

As demonstrated in recent studies and in our proposed model, such an integrated framework achieves better overall performance than approaches that treat forecasting and inventory optimization as separate modules. Specifically, leveraging causal knowledge strengthens the structural validity of forecasts, while aligning predictive outputs with inventory objectives ensures that decisions are both data-driven and economically efficient. This holistic approach is especially effective in settings with limited data or frequent distributional shifts, where conventional methods often struggle.

Table 2: Overview of studies on causal inference, forecasting, and inventory methods.

Source	**Causal Interference Methods**	**Forecasting Methods**	**Inventory Methods**
(Zhang et al., 2024)	Peter and Clark Momentary Conditional Independence	Ensemble Models	✗
(Sharma et al., 2024), (Fafoutellis and Vlahogianni, 2025)	Granger Causality	Deep Learning LSTM	✗
(Poletaev et al., 2024)	Granger Causality + Minimum Redundancy	Random Forest, XGBoost, LightGBM	✗
(Zhao et al., 2024)	Causal Feature Selection	CausalPatchTST	✗
(Yang et al., 2025), (Tian et al., 2024)	Convergent Cross Mapping	(LSTM, Bi-LSTM, CNNs, N-Beats, DLiner, Transformer) + (GBM + RF)	✗
(Kourentzes et al., 2020), (Babai et al., 2022)	✗	Exponential Smoothing	Service Level
(Tliche et al., 2020)	✗	WMA	Order up to policy
(Seyedan et al., 2023), (Sukolkit et al., 2024)	✗	Ensemble Models (LSTM)	Reorder point, Safety stock
(van der Haar et al., 2024)	✗	Ensemble Models	Optimal Order Quantity
Our proposed model	**Granger Causality**	**WTSMixer**	**Inventory cost**

3. Methodology

3.1. Overview

In this study, we propose WTSMixer, an enhanced time-series forecasting model that integrates Welford normalization into the TSMixer architecture. This approach improves forecasting accuracy and stability by

effectively handling data variability and outliers. Our methodology consists of four key steps:

- Data Preprocessing: To ensure data consistency and dependability, we gather and process data from several sources, addressing outliers and values that are missing with Welford normalization.
- Causal Inference: Using Augmented Dickey-Fuller tests, Granger causality tests, and Directed Acyclic Graphs (DAGs), we identify causal relationships between influential factors and demand fluctuations.
- Forecasting with WTSMixer: We extend the TSMixer model by incorporating Welford normalization at both input and output stages.To identify the temporal relationships and features interactions, the model employs time-mixing and features-mixing layers. A temporal projection layer is then used to produce more accurate forecasting results.
- Inventory Optimization: Forecasted demand guides inventory decisions by calculating stockout probability, safety stock levels, reorder points, and shortage costs, ultimately minimizing total inventory costs through an optimized replenishment policy.

By integrating Welford normalization into TSMixer, our proposed WTSMixer model enhances forecasting performance and adaptability to dynamic market conditions. Moreover, incorporating causal inference allows the model to leverage real-world causal relationships rather than relying solely on correlations, leading to more interpretable predictions and better-informed decision-making in inventory management. An overview of the proposed framework is shown in Figure 1.

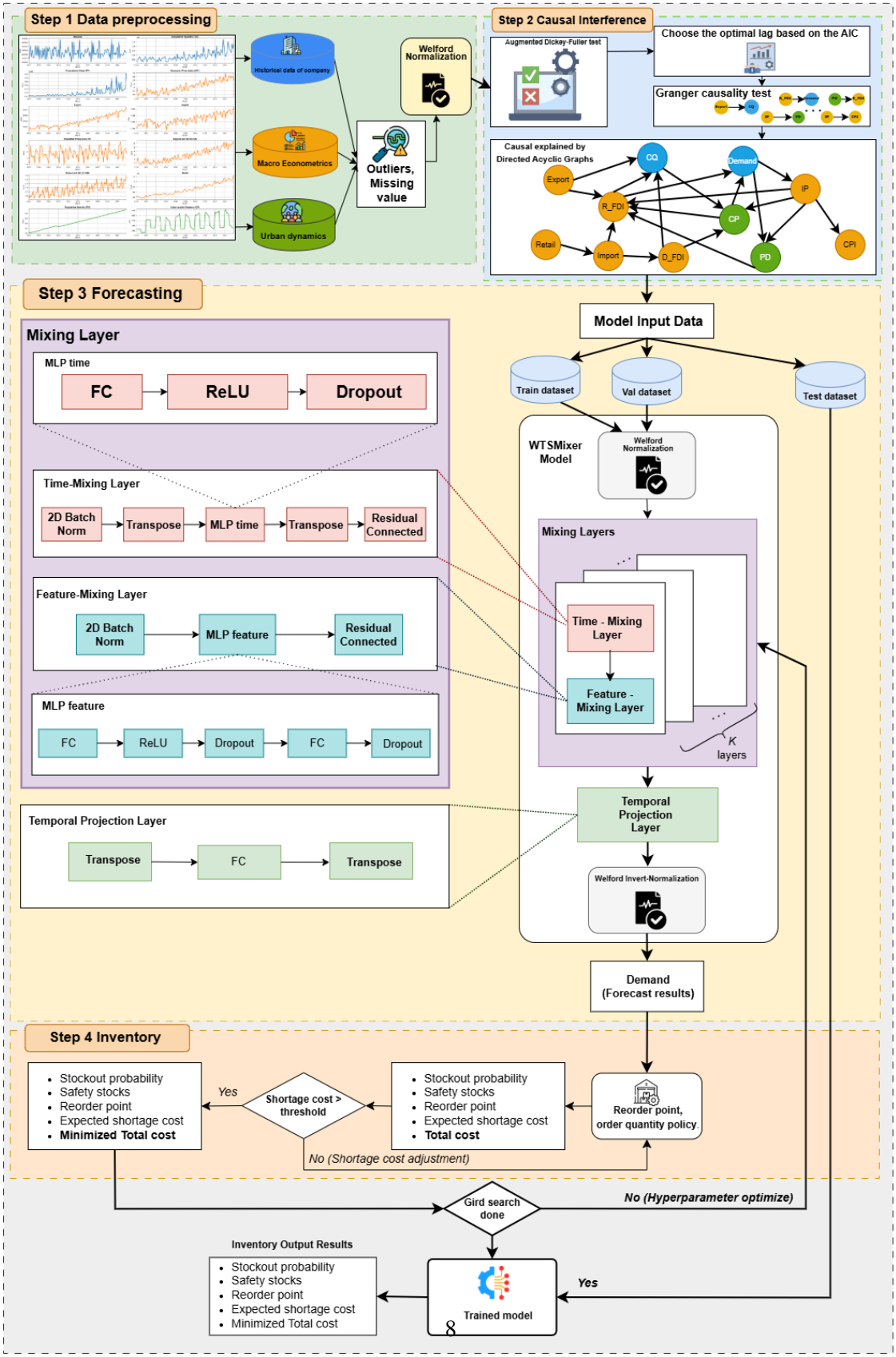

Figure 1: Proposed framework for demand forecasting and inventory optimization, integrating data preprocessing, causal inference, forecasting with the WTSMixer model, and inventory cost minimization.

3.2. Granger Causality

3.2.1. Definition of Granger Causality

Definition 1. *(Granger, 1969) Granger defines* $\mathbf{X}_{jt}$ *as a cause of* $\mathbf{X}_{it}$ *if:*

$$var(\mathbf{X}_{it} - \hat{\mathbf{X}}_{it} \mid \mathcal{H}_{<t}) < var(\mathbf{X}_{it} - \hat{\mathbf{X}}_{it} \mid \mathcal{H}_{<t} \setminus \mathbf{X}_{j,<t}), \tag{1}$$

where $\mathbf{X}_{it}$ *and* $\mathbf{X}_{jt}$ *denote time series i and j, respectively, which are assumed to be linear and stationary at time t. The symbol* $\mathcal{H}_{<t}$ *represents the information set containing all observations available prior to t, that is, up to* $t-1$*. The notation* $\hat{\mathbf{X}}_{it} \mid \mathcal{H}_{<t}$ *refers to the most accurate forecast of* $\mathbf{X}_{it}$ *given the information set* $\mathcal{H}_{<t}$*. Meanwhile,* $\mathbf{X}_{j,<t}$ *denotes the historical values of time series* $\mathbf{X}_{jt}$ *before time t, and* $\mathcal{H}_{<t} \setminus \mathbf{X}_{j,<t}$ *indicates the information set with the history of* $\mathbf{X}_{j,<t}$ *removed, retaining only the past of* $\mathbf{X}_{it}$*.*

3.2.2. Granger Causality Test

Understanding the difference between Granger causality and structural (true) causation is crucial: the latter concerns how an intervention on a driver produces a change in the outcome. By contrast, saying that $\mathbf{X}_{it}$ Granger-causes $\mathbf{X}_{jt}$ means only that lags of $\mathbf{X}_{it}$ contain incremental predictive information for forecast-horizon values of $\mathbf{X}_{jt}$. The Granger causality testing process is now explained (Poletaev et al., 2024).

To test whether $\mathbf{X}_{it}$ Granger-causes $\mathbf{X}_{jt}$, consider the hypotheses

- **H0**: $\mathbf{X}_{it}$ does not Granger-cause $\mathbf{X}_{jt}$.
- **H1**: $\mathbf{X}_{jt}$ is Granger-caused by $\mathbf{X}_{it}$.

If $\mathbf{X}_{i(t-k)}$ and $\mathbf{X}_{j(t-k)}$ are the respective k-th lags of $\mathbf{X}_{it}$ and $\mathbf{X}_{jt}$, Granger causality from $\mathbf{X}_{jt}$ to $\mathbf{X}_{it}$ is tested by comparing the predictive power of two models for $\mathbf{X}_{it}$: one using only its own past values, and another including past values of $\mathbf{X}_{jt}$ as additional predictors.

Restricted equation:

$$\mathbf{X}_{it} = \alpha_0 + \sum_{k=1}^{\ell} \alpha_k \mathbf{X}_{i(t-k)} + \epsilon_t, \tag{2}$$

where the lag length ℓ is selected based on an Akaike Information Criterion (AIC).

Unrestricted specification.

$$\mathbf{X}_{it} = \alpha_0 + \sum_{k=1}^{\ell} \alpha_k \, \mathbf{X}_{i(t-k)} + \sum_{k=p}^{q} \beta_k \, \mathbf{X}_{j(t-k)} + \epsilon_t, \tag{3}$$

where p and q delimit the lower and upper orders of the cross–lag window for $\mathbf{X}_{jt}$ (chosen based on statistical relevance), and ℓ is the maximal own–lag order for $\mathbf{X}_{it}$. The term α_0 is the intercept; the sequences $\{\alpha_k\}_{k=1}^{\ell}$ and $\{\beta_k\}_{k=p}^{q}$ weight, respectively, the lagged values of $\mathbf{X}_{i(t-k)}$ (own lags) and $\mathbf{X}_{j(t-k)}$ (cross lags).

The null of no Granger influence from X_{jt} to X_{it} is

$$H_0 : \beta_p = \cdots = \beta_q = 0, \qquad H_1 : \exists k \in \{p, \ldots, q\} \text{ with } \beta_k \neq 0.$$

These linear restrictions are evaluated with the conventional F statistic. Failure to reject H_0 implies that past realizations of $\mathbf{X}_{jt}$ add no predictive content for $\mathbf{X}_{it}$; rejection provides evidence of Granger causality from X_{jt} to X_{it}.

Specifically:

$$F = \frac{(\mathrm{RSS}_{\mathrm{red}} - \mathrm{RSS}_{\mathrm{full}})/(r - s)}{\mathrm{RSS}_{\mathrm{full}}/(T - r)}, \tag{4}$$

where $\mathrm{RSS}_{\mathrm{full}}$ denotes the residual sum of squares from the full (unrestricted) model, which includes r estimated parameters, while $\mathrm{RSS}_{\mathrm{red}}$ denotes the residual sum of squares from the reduced (restricted) model, which includes s estimated parameters. The symbol T represents the sample size, that is, the number of observations.

Hypothesis Testing

Null Hypothesis H_0: The time series $\mathbf{X}_{jt}$ does not Granger-cause the time series $\mathbf{X}_{it}$.

Alternative Hypothesis H_1: The time series $\mathbf{X}_{jt}$ Granger-causes the time series $\mathbf{X}_{it}$.

Testing the time series' stationarity is essential before performing Granger causality analysis. The ADF Test is a popular technique for assessing stationarity (Hu et al., 2025). Finding the model's ideal lag order d is an essential step in running the test. The ideal latency is determined using the Akaike Information Criterion (AIC) (Sharma et al., 2024).

3.2.3. Granger Causality Algorithm

Algorithm 1 describes the Granger Causality Testing Algorithm and explains the sequential procedures required to confirm stationarity, selecting the optimal lag length, conducting the Granger causality test, and visualizing the inferred causal relationships.

Algorithm 1: Granger Causality Testing Algorithm.

Input: Multivariate time series data $\mathcal{D}$, maximum lag L, significance level α.

Output: Causal relationships between variables in $\mathcal{D}$.

Function `GrangerCausality(`$\mathcal{D}, L, \alpha$`)`**:**
  foreach *pair of variables* $\left(\mathbf{X}_{it}, \mathbf{X}_{jt}\right)$ *in* $\mathcal{D}$ **do**
    if *StationarityTest*$(\mathbf{X}_{it})$ == *False* **then**
      $\mathbf{X}_{it} \leftarrow$ `MakeStationary(`$\mathbf{X}_{it}$`)` ; `// Transform` $\mathbf{X}_{it}$ `to stationary.`
    if *StationarityTest*$(\mathbf{X}_{jt})$ == *False* **then**
      $\mathbf{X}_{jt} \leftarrow$ `MakeStationary(`$\mathbf{X}_{jt}$`)` ; `// Transform` $\mathbf{X}_{jt}$ `to stationary.`
    $\ell^\star$ = `OptimalLagSelection(`$\mathbf{X}_{it}, \mathbf{X}_{jt}, L$`)` ; `// Select optimal lag using AIC.`
    p-value$_{it\to jt}$ = `GrangerTest(`$\mathbf{X}_{it}, \mathbf{X}_{jt}, \ell^\star$`)` ; `// Perform Granger causality test from` $\mathbf{X}_{it}$ `to` $\mathbf{X}_{jt}$`.`
    if p-*value*$_{it\to jt} < \alpha$ **then**
      `RecordCausality(`$\mathbf{X}_{it} \to \mathbf{X}_{jt}$`)` ; `// Store causal relationship from` $\mathbf{X}_{it}$ `to` $\mathbf{X}_{jt}$`.`
    p-value$_{jt\to it}$ = `GrangerTest(`$\mathbf{X}_{jt}, \mathbf{X}_{it}, \ell^\star$`)` ; `// Perform Granger causality test from` $\mathbf{X}_{jt}$ `to` $\mathbf{X}_{it}$`.`
    if p-*value*$_{jt\to it} < \alpha$ **then**
      `RecordCausality(`$\mathbf{X}_{jt} \to \mathbf{X}_{it}$`)` ; `// Store causal relationship from` $\mathbf{X}_{jt}$ `to` $\mathbf{X}_{it}$`.`
    if *both p-values are significant* **then**
      if *CausalityExists*$(\mathbf{X}_{it}, \mathbf{X}_{jt})$ **then**
        `Discard(`$\mathbf{X}_{it}, \mathbf{X}_{jt}$`)` ; `// Remove pair if both directions of causality exist.`
  return Causal graph $\mathcal{G}$;

3.3. WTSMixer

3.3.1. Forecasting Problem Formulation

This study adopts a three-month-ahead demand horizon to meet practical inventory requirements. The window provides sufficient lead time to adjust supply and production plans; reflects common practice in which three-month projections guide ordering, inventory control, and production scheduling; enables proactive inventory preparation and improves short-term forecast accuracy—reducing stockouts and overstock; and aligns with many industries' production and supply cycles, supporting process optimization and mitigating supply-chain risk.

To address the forecasting task, our goal is to predict a future sequence $\hat{\boldsymbol{\mathcal{X}}} \in \mathbb{R}^{N\times F}$ based on a given historical sequence $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{S\times C}$. The historical data is a sequence $\boldsymbol{\mathcal{X}} = \{\mathbf{X}_t\}_{t=1}^{S}$, containing S multi-variate observations, where each $\mathbf{X}_t \in \mathbb{R}^C$ is a vector of C variables at a point in time. The target sequence to be forecasted, $\hat{\boldsymbol{\mathcal{X}}} = \left\{\hat{\mathbf{X}}_t\right\}_{t=S+h}^{S+N+h}$, is composed of N future predictions, where each prediction $\hat{\mathbf{X}}_t$ contains F target variables. In this framework, S denotes the look-back window size, N is the length of the forecast horizon, and $h = 4$ signifies a 4-month lead time for the predictions. A formal description of how these input and

target sequences are constructed is provided in Equation (5).

$$\boldsymbol{\mathcal{X}} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_S \end{bmatrix} = \begin{bmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,C} \\ X_{2,1} & X_{2,2} & \dots & X_{2,C} \\ \vdots & \vdots & \ddots & \vdots \\ X_{S,1} & X_{S,2} & \dots & X_{S,C} \end{bmatrix}, \; \hat{\boldsymbol{\mathcal{X}}} = \begin{bmatrix} \hat{\mathbf{X}}_{S+h} \\ \hat{\mathbf{X}}_{S+h+1} \\ \vdots \\ \hat{\mathbf{X}}_{S+\mathcal{N}+h} \end{bmatrix} = \begin{bmatrix} \hat{X}_{S+h,1} & \hat{X}_{S+h,2} & \dots & \hat{X}_{S+h,F} \\ \hat{X}_{S+h+1,1} & \hat{X}_{S+h+1,2} & \dots & \hat{X}_{S+h+1,F} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{X}_{S+\mathcal{N}+h,1} & \hat{X}_{S+\mathcal{N}+h,2} & \dots & \hat{X}_{S+\mathcal{N}+h,F} \end{bmatrix}. \quad (5)$$

3.3.2. The Time-Series Mixer model (TSMixer)

TSMixer (Anonymous, 2026) is a multivariate forecasting model that replaces recurrent or convolutional modules with Multi-Layer Perceptrons (MLPs) to capture both temporal and cross-feature dependencies through dual mixing. The architecture consists of stacked Mixing Layers and a Temporal Projection Layer. Each Mixing Layer has two components: a Time-Mixing Layer and a Feature-Mixing Layer.

Given input tensor $\boldsymbol{\mathcal{A}}^{(d)} \in \mathbb{R}^{M \times T \times C}$, where M is batch size, T the look-back length, and C the number of variates.

Time-Mixing Layer. Begin with 2D batch normalization over batch and feature axes:

$$\boldsymbol{\mathcal{A}}^{(d)}_{t\_norm} = \texttt{2DBatchNorm}\left(\boldsymbol{\mathcal{A}}^{(d)}\right), \quad (6)$$

where $\boldsymbol{\mathcal{A}}^{(d)}$ is input tensor at layer d with dimensions (batch_size × time_steps × features), and `2DBatchNorm` is normalization applied across batch and feature dimensions.

Transpose time and feature axes to expose temporal patterns to the MLP:

$$\boldsymbol{\mathcal{A}}^{(d)}_{t\_trans1} = \texttt{Tensor\_Transpose}\left(\boldsymbol{\mathcal{A}}^{(d)}_{t\_norm}\right), \quad (7)$$

where `Tensor_Transpose` is axis permutation operation swapping time and feature dimensions.

Apply a temporal MLP with the sequence linear, ReLU, and dropout:

$$\boldsymbol{\mathcal{A}}^{(d)}_{t\_mlp} = \texttt{Dropout}\left(\texttt{ReLU}\left(\boldsymbol{\mathcal{A}}^{(d)}_{t\_trans1} * \mathbf{W}_1 \oplus \mathbf{b}_1\right), p\right), \quad (8)$$

where $\mathbf{W}_1$ is weight matrix for temporal expansion (time_steps × hidden_dim), $\mathbf{b}_1$ is corresponding bias vector, and p is dropout probability.

Project back to the original temporal size and transpose:

$$\boldsymbol{\mathcal{A}}^{(d)}_{t\_trans2} = \texttt{Tensor\_Transpose}\left(\boldsymbol{\mathcal{A}}^{(d)}_{t\_mlp} * \mathbf{W}_2 \oplus \mathbf{b}_2\right), \quad (9)$$

where $\mathbf{W}_2$ is projection weights (hidden_dim × time_steps), and $\mathbf{b}_2$ is output bias terms.

Add a residual connection to preserve the input:

$$\mathcal{A}_{time}^{(d)} = \mathcal{A}_{t\_trans2}^{(d)} + \mathcal{A}^{(d)}. \tag{10}$$

Feature-Mixing Layer

The Feature-Mixing Layer captures cross-variable dependencies at each time step. Given $\mathcal{A}_{\text{time}}^{(d)} \in \mathbb{R}^{M\times T\times C}$, it first applies 2D Batch Normalization:

$$\mathcal{A}_{f\_norm}^{(d)} = \texttt{2DBatchNorm}(\mathcal{A}_{time}^{(d)}). \tag{11}$$

A two-layer MLP transformation follows, including ReLU activation, dropout, and linear projection:

$$\mathcal{A}_{f\_mlp}^{(d)} = \texttt{Dropout}\left(\texttt{ReLU}\left(\mathcal{A}_{f\_norm}^{(d)} * \mathbf{W}_2 \oplus \mathbf{b}_2\right), p\right) * \mathbf{W}_3 \oplus \mathbf{b}_3. \tag{12}$$

A residual connection ensures stability and preserves information. Depending on feature dimensions F and C:

$$\mathcal{A}_{feature}^{(d)} = \begin{cases} \mathcal{A}_{f\_mlp}^{(d)} + \mathcal{A}_{time}^{(d)}, & \text{if } F = C, \\ \left(\mathcal{A}_{f\_mlp}^{(d)} * \mathbf{W}_4 \oplus \mathbf{b}_4\right) + \mathcal{A}_{time}^{(d)}, & \text{if } F < C. \end{cases} \tag{13}$$

Temporal Projection Layer

The Temporal Projection Layer serves to map the temporal dimension of an input. It effectively converts the sequence length from that of the look-back window, T, to the target forecast horizon's length, H. This layer operates on the output tensor from the final mixing block, $\mathcal{A}_{\text{mixing}}^{(d)} \in \mathbb{R}^{M\times T\times F}$. The transformation process begins with a transposition of the temporal and feature axes:

$$\mathcal{A}_{p\_trans}^{(d)} = \texttt{Tensor\_Transpose}(\mathcal{A}_{mixing}^{(d)}). \tag{14}$$

A fully connected (FC) layer projects the sequence to forecast length H:

$$\mathcal{A}_{p\_fc}^{(d)} = \mathcal{A}_{p\_trans}^{(d)} * \mathbf{W}_4 \oplus \mathbf{b}_4. \tag{15}$$

Finally, a second transpose restores the original shape, yielding the final output:

$$\hat{\mathcal{A}}^{(d)} = \texttt{Tensor\_Transpose}(\mathcal{A}_{p\_fc}^{(d)}) \in \mathbb{R}^{M\times H\times F}. \tag{16}$$

Thus, A Temporal Projection Layer and stacked Mixer Layers make up TSMixer, which efficiently projects to the forecast horizon while capturing temporal and feature dependencies.

Algorithm 2 presents the TSMixer Model algorithm, which describes the steps to take in order to process

the input tensor and produce the predicted output tensor.

Algorithm 2: TSMixer Model Algorithm.

Input: Input tensor $\mathcal{A}^{(d)}$; K, the total number of Mixing Layers in TSMixer.

Output: Predicted output tensor $\hat{\mathcal{A}}^{(d)}$.

Function `TSMixer(`$\mathcal{A}^{(d)}, K$`):`

- Initialize $\mathcal{A}_{1}^{(d)} = \mathcal{A}^{(d)}$; `// Initialization of the input tensor.`
- **for** $k = 1$ **to** K **do**
  - $\mathcal{A}_{time}^{(d)} =$ `Time_Mixing(`$\mathcal{A}_{k}^{(d)}$`)` ; `// Apply the Time-Mixing Layer.`
  - $\mathcal{A}_{feature}^{(d)} =$ `Feature_Mixing(`$\mathcal{A}_{time}^{(d)}$`)` ; `// Apply the Feature-Mixing Layer.`
  - Update $\mathcal{A}_{k+1}^{(d)} = \mathcal{A}_{feature}^{(d)}$; `// Pass the result to the next layer.`
- Set $\mathcal{A}_{mixing}^{(d)} = \mathcal{A}_{K}^{(d)}$; `// Final tensor after` K `Mixing Layers.`
- $\hat{\mathcal{A}}^{(d)} =$ `Projection(`$\mathcal{A}_{mixing}^{(d)}$`)` ; `// Apply the Temporal Projection Layer.`
- **return** $\hat{\mathcal{A}}^{(d)}$;

3.3.3. WTSMixer Model

The WTSMixer algorithm (Algorithm 3) couples Welford's online normalization with the TSMixer architecture for multivariate time-series forecasting. Welford's method provides stable, streaming normalization that adapts to distribution shifts; the normalized series is then processed by TSMixer to capture long-range temporal patterns; an inverse Welford step restores predictions to the original scale. This design enables real-time adaptability and improved accuracy across domains (e.g., finance, energy, logistics).

Algorithm 3: WTSMixer Model Algorithm.

Input: Tensor $\mathcal{X}^{(d)}$; K, number of Mixing Layers.

Output: Predicted output tensor $\hat{\mathcal{Y}}^{(d)}$.

Function `WTSMixer(`$\mathcal{X}^{(d)}, K$`):`

- $\mathcal{A}^{(d)} \leftarrow$ `Welford_Norm(`$\mathcal{X}^{(d)}$`)` ; `// Welford Normalization.`
- $\hat{\mathcal{A}}^{(d)} \leftarrow$ `TSMixer(`$\mathcal{A}^{(d)}, K$`);` `// Apply TSMixer model (Algorithm 2).`
- $\hat{\mathcal{Y}}^{(d)} \leftarrow$ `Welford_Invert(`$\hat{\mathcal{A}}^{(d)}$`)` ; `//Inverse Welford Normalization.`
- **return** $\hat{\mathcal{Y}}^{(d)}$;

3.4. Inventory Optimization Model

To effectively manage inventory in uncertain demand conditions, this study implements a probabilistic approach that integrates demand forecasting with inventory optimization. The model ensures adequate stock levels while minimizing associated costs.

3.4.1. The (r, q) Policy Under Uncertain Demand

Continuous review under the (r, q) policy offers an effective inventory-control scheme for variable demand. The reorder point r is the stock threshold that triggers replenishment, set from expected demand and its variability during lead time; the order quantity q is a constant lot placed at r that balances ordering and holding costs. The EOQ framework guides the cost-minimizing q, where total cost comprises ordering cost o (fixed per order: processing, transportation, handling) and holding cost h (monthly per unit: warehousing, insurance, opportunity cost).

Traditional EOQ models assume constant demand D, which is often unrealistic in practice. To address this limitation, we incorporate demand forecasting through the following adaptation:

$$\mu_D = \frac{1}{N} \sum_{i=1}^{N} F_i, \tag{17}$$

where μ_D represents the average forecasted demand over $\mathcal{N}$ months, and F_i is the forecasted demand for month i obtained from our WTSMixer model. This approach captures demand trends while smoothing out random fluctuations.

The optimal order quantity Q^* is then calculated as:

$$Q^* = \sqrt{\frac{2 \cdot \mu_D \cdot o}{h}}. \tag{18}$$

Equation (18) demonstrates the classic square root relationship in EOQ models, showing how higher demand or ordering costs increase the optimal order quantity, while higher holding costs decrease it. The square root nature indicates that order quantities increase at a decreasing rate with demand, highlighting the economies of scale in ordering.

3.4.2. Service-Level-Driven Replenishment

To manage the risk of stockouts, we introduce a service level γ representing the desired probability of meeting all demand during the lead time. The relationship between service level and stockout risk is expressed as:

$$\gamma = P(\boldsymbol{X} \leq r) = 1 - \frac{h \cdot Q^*}{c_{\text{shortage}} \cdot \mu_D}, \tag{19}$$

where c_{shortage} is the cost per unit short, reflecting the financial impact of lost sales or backorders. This equation establishes the critical trade-off between inventory holding costs and shortage costs.

The reorder point 20 and safety stock 21 are computed as follows:

$$r = \mu_D \cdot L + SS, \tag{20}$$

$$SS = z_\gamma \cdot \sigma_D \cdot \sqrt{L}, \tag{21}$$

where $z_\gamma = \Phi^{-1}(\gamma)$ represents the percent-point (inverse CDF) of the standard normal at γ, matching the required service level, σ_D represents demand variability (standard deviation), and L is the lead time in months.

Equation (21) shows that safety stock increases with higher demand variability (σ_D) and longer lead times (L), but scales with the square root of the lead time due to statistical pooling effects. The reorder point in Equation (20) combines the expected demand during lead time ($\mu_D \cdot L$) with the safety buffer (SS).

3.4.3. Cost Optimization Framework

The total inventory cost (TC) consists of three key components:

- **Ordering cost**:

$$c_{\text{ordering}} = \frac{\mu_D}{Q^*} \cdot o. \tag{22}$$

  Equation (22) represents the annual ordering cost, where $\frac{\mu_D}{Q^*}$ gives the number of orders placed per year. The cost decreases with larger order quantities due to fewer orders being placed.

- **Holding cost**:

$$c_{\text{holding}} = \left(\frac{Q^*}{2} + \max(SS, 0) \right) \cdot h. \tag{23}$$

  The first term $\frac{Q^*}{2}$ reflects the average cycle stock (half of order quantity), while the second term accounts for safety stock. The $\max(SS, 0)$ function ensures we only consider positive safety stock levels.

- **Shortage cost** is derived through the loss function approach:

$$\mathcal{L}(v) = \int_v^\infty (u - v) \cdot \phi(u) \cdot du, \tag{24}$$

$$\mathbb{E}(\text{Shortage}) = \mathcal{L}(v) \cdot \sigma_D \cdot \sqrt{L}, \tag{25}$$

$$\mathbb{E}(c_{\text{shortage}}) = \frac{c_{\text{shortage}} \cdot \mu_D \cdot \mathbb{E}(\text{Shortage})}{Q^*}. \tag{26}$$

  The loss function $\mathcal{L}(v)$ in Equation (24) calculates the expected number of standard deviations by which demand exceeds available inventory. Equation (25) scales this to actual units using demand

variability and lead time. Finally, Equation (26) annualizes the shortage cost by considering the frequency of orders ($\frac{\mu_D}{Q^*}$).

The comprehensive **total cost** function combines all components:

$$TC = c_{\text{ordering}} + c_{\text{holding}} + \mathbb{E}(c_{\text{shortage}}). \tag{27}$$

This integrated approach jointly optimizes order quantity (Q^*), reorder point (r), and safety stock (SS), explicitly balancing ordering efficiency, holding costs, and service quality. Its probabilistic foundation enables robust control under realistic demand uncertainty, delivering both cost efficiency and reliable customer service.

4. Experiments

Penalty cost is used to quantify the impact of demand-forecast errors and tune inventory policy to minimize shortages, optimize stock levels, and reduce related costs. It represents the financial effect of unmet demand, with the penalty rate μ and product price P specified by the business to reflect operational and financial realities; these parameters are calibrated to the company's context to capture the cost of lost sales accurately. The proposed model integrates a dedicated forecasting method with an inventory control strategy and is evaluated under the three scenarios in Section 3.1; the experimental workflow is summarized in Figure 2.

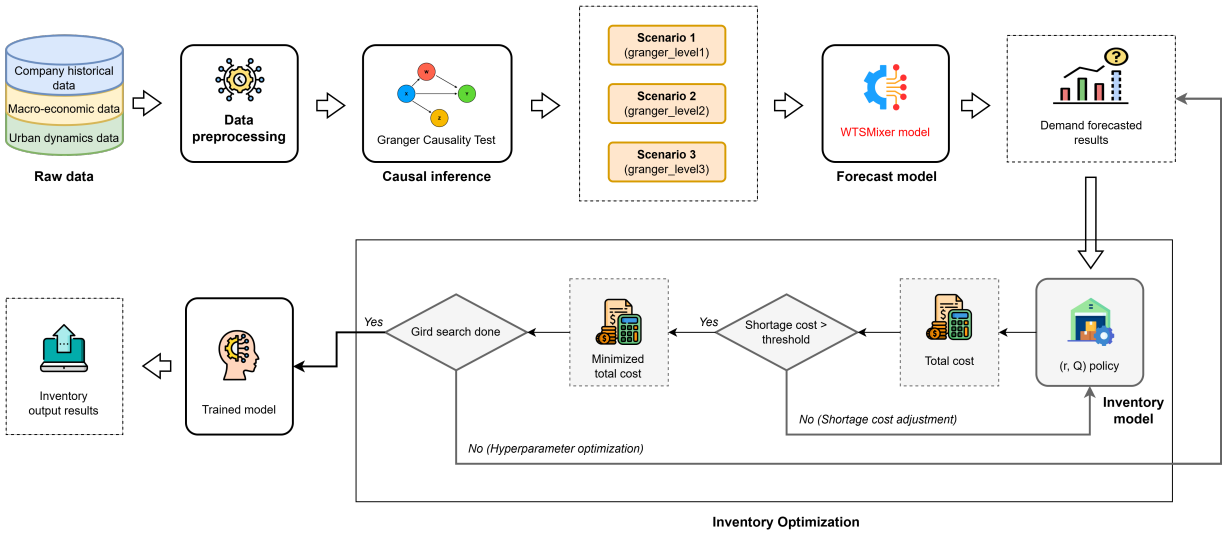

Figure 2: Three scenarios are used in the experimental method.

4.1. Dataset

This study forecasts demand for Product B using historical operational data from a logistics company (2015–2024), complemented by macroeconomic indicators from Vietstock[1], urban dynamics from the Ministry of Construction[2] (construction projects) and the General Statistics Office[3] (population density).

The feature set comprises three groups: (1) *historical company data*—Demand (daily sales aggregated monthly), Competitor Quantity (CQ; monthly competitor sales), and Promotional Fund (PF; monthly promotional expenditure; cf. Abolghasemi et al. (2020)); (2) *macro econometrics*—indicators listed in Table 3; and (3) *urban dynamics*—Population Density (PD; average people per km^2) and Construction Projects (CP; civil-project counts).

Table 3: Macroeconomic data collected from Vietstock.

ID	Attribute Name	Description	Unit
1	Consumer Price Index (CPI)	Index reflecting relative changes in consumer goods prices over time.	%
2	Export	Total value of goods exported, calculated at FOB prices.	Million USD
3	Import	Total value of goods imported, calculated at CIF prices.	Million USD
4	Industrial Production (IP)	Ratio of industrial production volume compared to a base period.	%
5	Registered FDI (R-FDI)	Total registered foreign direct investment projects in a given period.	Million USD
6	Disbursed FDI (D-FDI)	Actual FDI disbursed as realized investment capital.	Million USD
7	Retail	Total value of goods and services sold to consumers.	Billion VND

All datasets were collected and aggregated over a three-year period from 2015 to 2024 on a monthly basis to support the forecasting of Product B demand for the next 12 months. The research dataset consists of 120 rows and 12 columns. This dataset is detailed in Figure 3, showcasing the data used to forecast the demand for Product B based on these various attributes.

[1]Vietstock sources: Vietstock website
[2]Ministry of Construction website: `ConstructionData`
[3]General Statistics Office website: `https://www.gso.gov.vn/dan-so/`

Figure 3: Time Series Visualization of Dataset Attributes.

4.2. Granger Causality

Figure 4 summarizes the inferred Granger causal relationships among the time-indexed variables. For forecasting "Demand", first-order causes are R_FDI and CP; second-order causes are Export, Import, IP, PD, CQ, and D_FDI; third-order involves Retail. Accordingly, three evaluation scenarios are defined: S_1 = {Demand, R_FDI, CP} (3 series); $S_2 = S_1 \cup$ {Export, Import, IP, PD, CQ, D_FDI} (9 series); and $S_3 = S_2 \cup$ {Retail} (10 series).

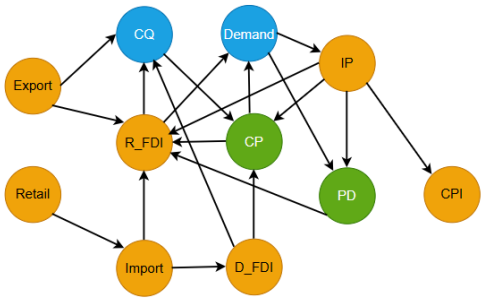

Figure 4: Granger Causal Inference Results.

4.3. Baselines

Performance is evaluated against established and state-of-the-art sequence models—Long Short-Term Memory (LSTM), Deep Neural Networks (DNN), and Gated Recurrent Units (GRU) on real datasets. These baselines enable unbiased, statistically grounded comparisons that highlight improvements in the proposed model's generalizability and predictive accuracy. Beyond serving as reference points, they also evidence feasibility in real-world inventory management. The experimental assessment is designed to show that the proposed approach not only improves forecasting accuracy but also materially supports superior inventory decision-making.

4.4. Hyperparameter Optimization

To find the best model configuration for the prediction length $H = \{3\}$, we perform a hyperparameter optimization process using the Grid Search method. This process investigates a predefined set of values for the core hyperparameters: `sequence_length`, `num_blocks`, `dropout_rate`, `samples_per_batch`, `hidden_layer_dim`, and `learning_rate`. The objective is to find the combination that minimizes efficiency metrics like Total Cost (Belete and Huchaiah, 2022). The optimal hyperparameters resulting from this process, as listed in Table 4, are selected to train the final model.

Scenario 1. GrangerL1_data								
Model	**predict_length**	**sequence_length**	**num_blocks**	**dropout_rate**	**samples_per_batch**	**hidden_layer_dim**	**learning_rate**	**optimal_total_cost**
	H	1,2,3...,6	1,2,...,4	0.1, 0.3, 0.5, 0.7, 0.9	1, 2, 3, 4	8, 16, 32, 64, 128, 256	1e-5, 1e-4	
WTSMixer	3	6	3	0.1	4	256	1e-4	304,763.09
TSMixer	3	6	3	0.1	4	256	1e-4	349,032.53
Scenario 2. GrangerL2_data								
Model	**predict_length**	**sequence_length**	**num_blocks**	**dropout_rate**	**samples_per_batch**	**hidden_layer_dim**	**learning_rate**	**optimal_total_cost**
	H	1,2,3...,6	1,2,...,4	0.1, 0.3, 0.5, 0.7, 0.9	1, 2, 3, 4	8, 16, 32, 64, 128, 256	1e-5, 1e-4	
WTSMixer	3	6	3	0.1	4	256	1e-4	336,929.75
TSMixer	3	6	3	0.1	4	256	1e-4	385,106.35
Scenario 3. GrangerL3_data								
Model	**predict_length**	**sequence_length**	**num_blocks**	**dropout_rate**	**samples_per_batch**	**hidden_layer_dim**	**learning_rate**	**optimal_total_cost**
	H	1,2,3...,6	1,2,...,4	0.1, 0.3, 0.5, 0.7, 0.9	1, 2, 3, 4	8, 16, 32, 64, 128, 256	1e-5, 1e-4	
WTSMixer	3	6	4	0.1	4	256	1e-4	339,145.41
TSMixer	3	6	4	0.1	4	256	1e-4	403,766.22

Table 4: The optimal hyperparameter configuration for every model and the hyperparameter space of search.

4.5. Evaluation Metrics

Assessing model performance is essential in time-series forecasting to verify the accuracy and reliability of its predictions; common measures include MAE, MSE, MAPE and RMSE. Formal definitions are given in Equation (28):

$$\begin{aligned} \text{MAE} &= \frac{1}{m}\sum_{i=1}^{m}\left|Y_i - \hat{Y}_i\right|, & \text{MSE} &= \frac{1}{m}\sum_{i=1}^{m}(Y_i - \hat{Y}_i)^2, \\ \text{MAPE} &= \frac{1}{m}\sum_{i=1}^{m}\frac{\left|Y_i - \hat{Y}_i\right|}{|Y_i|} \times 100, & \text{RMSE} &= \sqrt{\text{MSE}}. \end{aligned} \tag{28}$$

where m denotes the total number of data points in the dataset, Y_i represents the i-th sample's actual value, and $\hat{Y}_i$ represents the value that was predicted for the i-th sample.

While MAE, MSE, MAPE, and RMSE assess forecast accuracy, they often miss the business impact of errors. In Scenario 2, the Total Cost metric (Eq. 27) is introduced to directly quantify the economic consequences of predictions, linking accuracy to resource allocation and inventory decisions.

4.6. Results

Empirical results in Tables 5–7 show that WTSMixer consistently delivers the best inventory optimization across all scenarios, achieving the lowest total costs (304,763.09; 336,929.75; 339,145.41) and a stable optimal shortage cost $c_s^* \approx 1.49$ (close to the true 1.53). LSTM and GRU yield competitive $TC_{\min}$ but with larger c_s^* variability (1.36–1.60), while TSMixer and DNN perform worse on both $TC_{\min}$ and c_s^* stability. Notably, despite some models showing better forecast metrics (MSE/MAE/MAPE), WTSMixer remains superior in inventory cost minimization. All ML models outperform the benchmark ($TC_{\min}$ = 525,533.42); WTSMixer reduces cost by up to 42%.

Table 5: Performance Assessment of All Models Against Ground Truth in Scenario 1
(In each row, **bold** marks the optimal value, and an underlined denotes the second-best).

Model	**MAE**	**MSE**	**MAPE (%)**	**RMSE**	$\boldsymbol{TC_{min}}$	$\boldsymbol{c_s^*}$
WTSMixer	84,531	12,860,997,053	27.516	113,406	**304,763.09**	1.49
TSMixer	82,901	13,597,517,776	22.631	116,608	349,032.53	1.68
LSTM	81,628	12,591,223,146	24.439	112,211	320,606.01	1.55
GRU	80,249	11,069,393,678	24.872	105,211	351,612.30	1.60
DNN	66,829	10,428,630,788	**21.968**	102,121	366,850.90	1.51
Real demand-based standard	-	-	-	-	525,533.42	1.53

Table 6: Performance Assessment of All Models Against Ground Truth in Scenario 2
(In each row, **bold** marks the optimal value, and an underlined denotes the second-best).

Model	MAE	MSE	MAPE (%)	RMSE	TC_{min}	c_s^*
WTSMixer	83,709	12,181,440,490	26.973	110,370	**336,929.75**	1.49
TSMixer	101,913	16,632,009,599	29.085	128,965	385,106.35	1.72
LSTM	121,834	17,401,284,923	42.149	131,914	356,999.63	1.37
GRU	71,048	10,251,750,051	**23.162**	101,251	371,403.96	1.57
DNN	124,117	20,660,124,280	43.066	143,736	368,605.23	1.38
Real demand-based standard	-	-	-	-	525,533.42	1.53

Table 7: Performance Assessment of All Models Against Ground Truth in Scenario 3
(In each row, **bold** marks the optimal value, and an underlined denotes the second-best).

Model	MAE	MSE	MAPE (%)	RMSE	TC_{min}	c_s^*
WTSMixer	87,657	12,151,839,726	28.698	110,235	**339,145.41**	1.48
TSMixer	122,620	18,414,309,118	40.768	135,699	403,766.22	1.39
LSTM	123,940	18,192,506,990	43.337	134,880	355,930.89	1.36
GRU	76,357	11,970,124,150	**24.356**	109,408	367,688.46	1.59
DNN	148,047	26,730,702,312	52.030	163,495	373,526.20	1.34
Real demand-based standard	-	-	-	-	525,533.42	1.53

As shown in Figures 5–7, WTSMixer exhibits the lowest median and tightest total-cost dispersion, consistent with Tables 5–7. Across Scenarios 1–3, $TC_{\min}$ = 304,763.09; 336,929.75; 339,145.41—approximately ≈ 4.9%, ≈ 5.6%, and ≈ 4.7% below the second-best (LSTM). The shortage-cost estimate is stable at $c_s^* \approx 1.49$, close to the ground truth 1.53, indicating reliable threshold identification. Although LSTM/GRU achieve competitive $TC_{\min}$, their c_s^* varies more (1.36–1.60), while TSMixer/DNN underperform on both $TC_{\min}$ and c_s^* stability. Overall, inventory-cost optimization depends not only on forecast accuracy but also on explicitly modeling operational parameters—especially shortage cost.

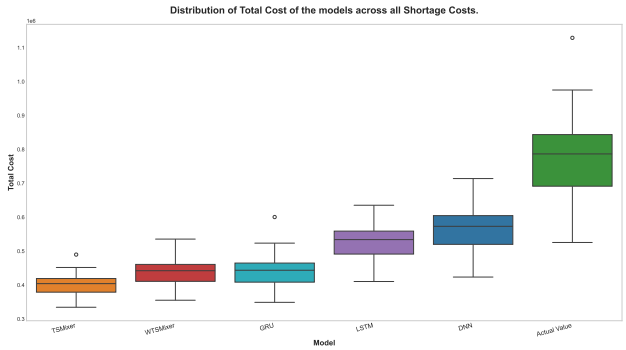

Figure 5: Total cost across models with different shortfall costs per unit (Scenario 1).

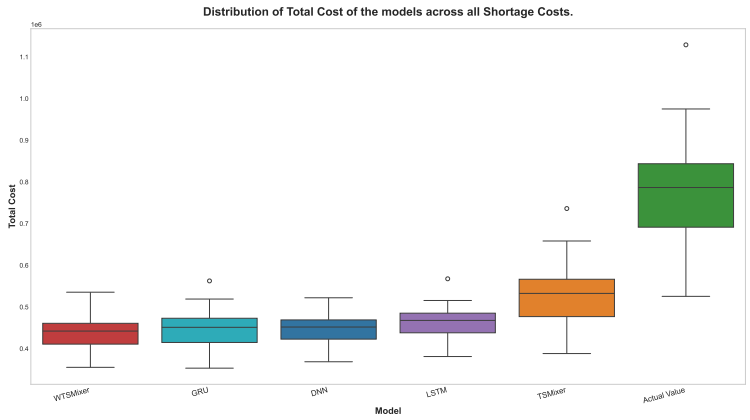

Figure 6: Total cost across models with different shortfall costs per unit (Scenario 2).

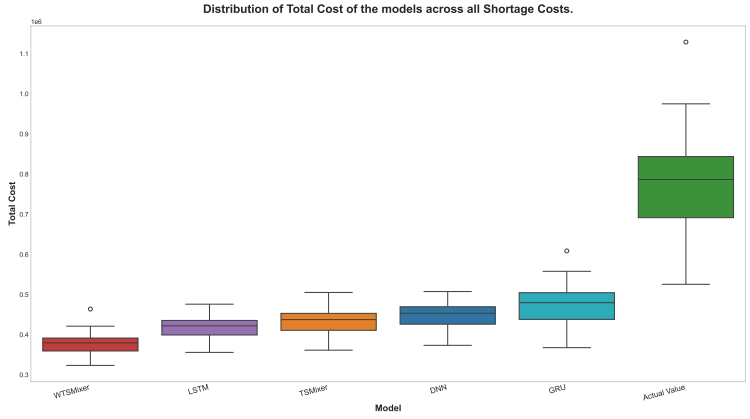

Figure 7: Total cost across models with different shortfall costs per unit (Scenario 3).

5. Conclusion and discussion

This study proposes a unified, causality-aware framework that links multivariate time-series forecasting with inventory optimization for dynamic e-commerce. The approach integrates Granger-causality–based feature selection, a WTSMixer deep architecture, and cost-sensitive modeling under the (r, Q) policy, aligning predictive outputs with operational decisions. The work advances causal inference, demand forecasting, and inventory performance via a combined theoretical–empirical analysis for uncertain environments, with all components—Granger causality, the WTSMixer-based TSMixer, and a probabilistic inventory-cost model—coherently integrated and documented for transparency and reproducibility.

WTSMixer delivers substantial improvements on both forecast- and cost-based metrics across three scenarios with Granger L1–L3 input selection: it surpasses standard TSMixer in accuracy, achieves the lowest total inventory cost in every scenario, and outperforms LSTM, GRU, and DNN on inventory outcomes.

Key innovations:

- **Welford normalization**: streaming, stable normalization for nonstationary multivariate series, mitigating distribution shifts and improving short-/long-horizon accuracy.
- **Dual objectives**: joint optimization of predictive fidelity and inventory cost, increasing cost-efficiency while preserving responsiveness.
- **Causally informed forecasts**: tighter safety-stock and reorder-point estimates, reducing stockouts and overstock.

Using a probabilistic inventory-cost model, the analysis quantifies how forecast accuracy shapes stockout probability, reorder points, expected shortage cost, and total cost, yielding managerial insights: (1) shortage cost is the dominant driver—higher shortage penalties prompt higher safety stock and r; (2) accuracy remains important but its impact is most evident under cost-optimized settings, where WTSMixer can outperform baselines even with similar or lower error; (3) under high shortage costs with error-only optimization, model differences narrow (strategy convergence), whereas under cost-based optimization, model choice materially affects total cost.

Overall, the results underscore the value of forecasting models that align statistical accuracy with downstream operational and financial objectives.

Future directions: extend TSMixer with architectural refinements (e.g., dynamic attention, hybrid feature encoders); scale to larger, multimodal datasets (text, images, high-frequency signals); automate hyperparameter tuning; and integrate with decision agents (e.g., reinforcement-learning inventory controllers) to enhance adaptability and automation, advancing robust, flexible, causally aligned supply-chain analytics.

CRediT authorship contribution statement

Nguyen Thi Ngoc Anh: Conceptualization; Methodology; Supervision; Project administration; Writing – original draft; Writing – review & editing; Funding acquisition. **Nguyen The Phong**: Methodology; Software; Data curation; Validation; Formal analysis; Visualization; Writing – review & editing. **Nguyen Thi Ha**: Data curation; Software; Investigation; Validation; Formal analysis; Visualization; Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work forms part of project BKFintech-2024.02 and was made possible by the generous support of Rikkeisoft Company and BK Fintech, HUST. The authors thank both institutions for their continued

support.

References

Abolghasemi, M., Beh, E., Tarr, G., and Gerlach, R. (2020). Demand forecasting in supply chain: The impact of demand volatility in the presence of promotion. *Computers & Industrial Engineering*, 142:106380.

Anonymous (2026). Details omitted for double-blind reviewing. Details omitted for double-blind reviewing.

Babai, M. Z., Dai, Y., Li, Q., Syntetos, A., and Wang, X. (2022). Forecasting of lead-time demand variance: Implications for safety stock calculations. *European Journal of Operational Research*, 296(3):846–861. `https://doi.org/10.1016/j.ejor.2021.04.017`.

Belete, D. M. and Huchaiah, M. D. (2022). Grid search in hyperparameter optimization of machine learning models for prediction of hiv/aids test results. *International Journal of Computers and Applications*, 44(9):875–886. `https://doi.org/10.1080/1206212X.2021.1974663`.

Benkő, Z., Zlatniczki, Á., Stippinger, M., Fabó, D., Sólyom, A., Erőss, L., Telcs, A., and Somogyvári, Z. (2024). Bayesian inference of causal relations between dynamical systems. *Chaos, Solitons & Fractals*, 185:115142.

Bodendorf, F., Sauter, M., and Franke, J. (2023). A mixed methods approach to analyze and predict supply disruptions by combining causal inference and deep learning. *International Journal of Production Economics*, 256:108708.

Chu, Z., Ding, H., Zeng, G., Wang, S., and Li, Y. (2024). Causal interventional prediction system for robust and explainable effect forecasting. *arXiv preprint arXiv:2407.19688*.

Coroneo, L. and Iacone, F. (2025). Testing for equal predictive accuracy with strong dependence. *International journal of forecasting*, 41(3):1073–1092.

Fafoutellis, P. and Vlahogianni, E. I. (2025). A theory-informed multivariate causal framework for trustworthy short-term urban traffic forecasting. *Transportation Research Part C: Emerging Technologies*, 170:104945.

Farsi, B., Amayri, M., Bouguila, N., and Eicker, U. (2021). On short-term load forecasting using machine learning techniques and a novel parallel deep lstm-cnn approach. *IEEE access*, 9:31191–31212.

Fildes, R., Ma, S., and Kolassa, S. (2022). Retail forecasting: Research and practice. *International Journal of Forecasting*, 38(4):1283–1318.

Gao, C., Zheng, Y., Wang, W., Feng, F., He, X., and Li, Y. (2024). Causal inference in recommender systems: A survey and future directions. *ACM Transactions on Information Systems*, 42(4):1–32.

Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438.

Guo, J., Zhang, Y., Xu, Q., Jiang, L., Liu, X., Lv, S., and Zhu, J. (2024). An robust n-gram causal inference approach based on multi-model fusion. *Physical Communication*, 64:102293.

Hu, J., Bai, J., Yang, J., and Lee, J. J. (2025). Crash risk prediction using sparse collision data: Granger causal inference and graph convolutional network approaches. *Expert Systems with Applications*, 259:125315.

Kourentzes, N., Trapero, J. R., and Barrow, D. K. (2020). Optimising forecasting models for inventory planning. *International Journal of Production Economics*, 225:107597. https://doi.org/10.1016/j.ijpe.2019.107597.

Lim, B., Arık, S. Ö., Loeff, N., and Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764.

Liu, Z., Xu, X., Luo, B., Yang, C., Gui, W., and Dubljevic, S. (2024). Accelerated mpc: A real-time model predictive control acceleration method based on tsmixer and 2d block stochastic configuration network imitative controller. *Chemical Engineering Research and Design*, 208:837–852.

Mitra, A., Jain, A., Kishore, A., and Kumar, P. (2022). A comparative study of demand forecasting models for a multi-channel retail company: a novel hybrid machine learning approach. In *Operations research forum*, volume 3, page 58. Springer.

Mrabet, Z., Alsamara, M., Mimouni, K., and Awwad, A. (2025). Do supply chain pressures affect consumer prices in major economies? new evidence from time-varying causality analysis. *Economic Modelling*, 142:106914.

Nitish, K. and Indu, J. (2025). Evaluating interdependencies of lake water surface temperature and clarity. *Science of The Total Environment*, 966:178695.

Oreshkin, B. N., Dudek, G., Pełka, P., and Turkina, E. (2021). N-beats neural network for mid-term electricity load forecasting. *Applied Energy*, 293:116918.

Poletaev, A., Liu, B., Li, L., Avagyan, V., and Voskanyan, V. (2024). Improve the prediction in the digital era: Causal feature selection with minimum redundancy. *Journal of Digital Economy*, 3:14–36.

Raifu, I. A., Obaniyi, F. A., Nnamani, G., and Salihu, A. A. (2025). Revisiting causal relationship between renewable energy and economic growth in oecd countries: Evidence from a novel jks's granger non-causality test. *Renewable Energy*, page 122559.

Sengupta, S., Chakraborty, T., and Singh, S. K. (2025). Forecasting cpi inflation under economic policy and geopolitical uncertainties. *International Journal of Forecasting*, 41(3):953–981.

Seyedan, M., Mafakheri, F., and Wang, C. (2023). Order-up-to-level inventory optimization model using time-series demand forecasting with ensemble deep learning. *Supply Chain Analytics*, 3:100024. `https://doi.org/10.1016/j.sca.2023.100024`.

Sharma, K., Dwivedi, Y. K., and Metri, B. (2024). Incorporating causality in energy consumption forecasting using deep neural networks. *Annals of Operations Research*, 339(1):537–572.

Sohrabpour, V., Oghazi, P., Toorajipour, R., and Nazarpour, A. (2021). Export sales forecasting using artificial intelligence. *Technological Forecasting and Social Change*, 163:120480.

Souto, H. G. (2024). Charting new avenues in financial forecasting with timesnet: The impact of intraperiod and interperiod variations on realized volatility prediction. *Expert Systems with Applications*, 255:124851.

Sukolkit, N., Arunyanart, S., and Apichottanakul, A. (2024). An open innovative inventory management based demand forecasting approach for the steel industry. *Journal of Open Innovation: Technology, Market, and Complexity*, page 100407. `https://doi.org/10.1016/j.joitmc.2024.100407`.

Tian, J., Wang, G., Huang, S., Xiang, D., and Li, W. (2024). Synergizing convergent cross-mapping and machine learning for reliable daily forecasting of riverine chlorophyll-a concentration. *Journal of Hydrology*, 645:132072.

Tliche, Y., Taghipour, A., and Canel-Depitre, B. (2020). An improved forecasting approach to reduce inventory levels in decentralized supply chains. *European Journal of Operational Research*, 287(2):511–527. `https://doi.org/10.1016/j.ejor.2020.04.044`.

van der Haar, J. F., Wellens, A. P., Boute, R. N., and Basten, R. J. (2024). Supervised learning for integrated forecasting and inventory control. *European Journal of Operational Research*, 319(2):573–586. `https://dx.doi.org/10.1016/j.ejor.2024.07.004`.

Virbickaitė, A., Lopes, H. F., and Zaharieva, M. D. (2025). Multivariate dynamic mixed-frequency density pooling for financial forecasting. *International Journal of Forecasting*, 41(3):1184–1198.

Wang, J., Chong, W. K., Lin, J., and Hedenstierna, C. P. T. (2024). Retail demand forecasting using spatial-temporal gradient boosting methods. *Journal of Computer Information Systems*, 64(5):652–664.

Yang, Y., Song, W., Han, S., Yan, J., Wang, H., Dai, Q., Huo, X., and Liu, Y. (2025). Power forecasting method of ultra-short-term wind power cluster based on the convergence cross mapping algorithm. *Global Energy Interconnection*.

Zhang, C., Nong, X., Behzadian, K., Campos, L. C., Chen, L., and Shao, D. (2024). A new framework for water quality forecasting coupling causal inference, time-frequency analysis and uncertainty quantification. *Journal of Environmental Management*, 350:119613.

Zhang, X., Hu, Y., Xie, K., Wang, S., Ngai, E., and Liu, M. (2014). A causal feature selection algorithm for stock prediction modeling. *Neurocomputing*, 142:48–59.

Zhao, H., Xu, P., Gao, T., Zhang, J. J., Xu, J., and Gao, D. W. (2024). Cptcfs: Causalpatchtst incorporated causal feature selection model for short-term wind power forecasting of newly built wind farms. *International Journal of Electrical Power & Energy Systems*, 160:110059.

Zhou, C., Li, H., Hu, Y., Zhang, B., Ren, P., Kan, Z., Jia, X., Mi, J., and Guo, X. (2025). Causal effects of key air pollutants and meteorology on ischemic stroke onset: A convergent cross-mapping approach. *Ecotoxicology and Environmental Safety*,

291:117861.

Appendix A. Nomenclatures and Notations

Table A.8: Glossary of abbreviations and acronyms used in this study.

Acronym	Full Term / Description	Acronym	Full Term / Description
Machine Learning & Deep Learning Models			
TSMixer	Time-Series Mixer	LightGBM	Light Gradient Boosting Machine
RNN	Recurrent Neural Network	SVR	Support Vector Regression
LSTM	Long Short-Term Memory	CNN	Convolutional Neural Network
GRU	Gated Recurrent Unit		
Statistical & Econometric Models			
ES	Exponential Smoothing	VAR	Vector Autoregressive
SES	Single Exponential Smoothing	ARIMA	Autoregressive Integrated Moving Average
WMA	Weighted Moving Average	SARIMA	Seasonal Autoregressive Integrated Moving Average
Model Components & Techniques			
RevIN	Reversible Instance Normalization	ReLU	Rectified Linear Unit
FC	Fully Connected		
Performance Metrics			
MSE	Mean Squared Error	RMSE	Root Mean Squared Error
MAE	Mean Absolute Error	MAPE	Mean Absolute Percentage Error
MMSE	Minimum Mean Squared Error		
Economic Indicators & Factors			
FDI	Foreign Direct Investment	CPI	Consumer Price Index
R-FDI	Registered FDI	IP	Industrial Production
D-FDI	Disbursed FDI	CP	Construction Projects
Inventory & Domain-Specific Terms			
DDI	Downstream Demand Inference	CQ	Competitor Quantity
EOQ	Economic Order Quantity		

Table A.9: Comprehensive glossary of symbols, variables, and parameters used throughout the study.

Symbol	Definition	Symbol	Definition
I. Data Structures & Dimensions			
$\mathcal{X}$	The input sequence, representing historical time-series data	$\hat{\mathcal{X}}$	The output sequence, representing future forecasted data
S_{len}	The number of time steps in a single input instance	N_{out}	The number of time steps in the forecast horizon
C	The dimensionality (number of features) of the input data	F	The dimensionality of the output data ($F \leq C$)
$\mathbf{X}_t$	A vector of all feature values at a past time step t	$\hat{\mathbf{X}}_t$	A vector of all predicted feature values at a future time step t
$\mathbf{X}_{it}$	The value of the target feature i at time step t	$\mathbf{X}_{jt}$	The value of a contextual feature j at time step t ($j \neq i$)
II. Inventory Management System			
Variables		*Constants*	
Q	The determined quantity for a replenishment order	H	The cost to hold a single unit in inventory for one month
Q^*	The optimal order quantity that minimizes total costs	S_{cost}	The fixed administrative cost associated with placing one order
D	The average demand observed per month	L	The lead time, in months, between placing and receiving an order
D_i	The actual, observed demand during month i	P	The purchase price per unit of an item
F_i	The forecasted demand for the upcoming month i	η	The rate used to calculate the penalty for stockouts
SS	The level of safety stock maintained to buffer against uncertainty	TC	The total relevant cost, combining both inventory and stockout costs
C_I	The total cost incurred from holding inventory	C_P	The total penalty cost resulting from stockouts
N_{months}	The total duration of the evaluation period, in months		
III. Model Configuration & Hyperparameters			
K	The depth of the model, defined by the number of mixing blocks	M	The number of instances processed in a single batch (batch size)

Table A.9 – continued from previous page

Symbol	Definition	Symbol	Definition
p	The probability of dropping out a neuron during training	learning_rate	The step size used by the optimizer to update model weights
hidden_layer_dim	The number of neurons in the hidden layers of the MLP blocks		

IV. Internal Architectural Components

Tensors		*Learnable Parameters & Operations*	
$\mathcal{X}^{(d)}$	The initial tensor fed into the model's input layer	$\mathbf{W}_1$	The weight matrix within the time-mixing MLP
$\hat{\mathcal{Y}}^{(d)}$	The final tensor produced by the model after all transformations	$\mathbf{W}_{2..4}$	The set of weight matrices within the feature-mixing MLP
$\mathcal{A}^{(d)}$	The tensor serving as input to a WTSMixer block	$\mathbf{W}_5$	The weight matrix for the final temporal projection layer
$\mathcal{A}^{(d)}_{time}$	The output representation from the Time-Mixing sub-layer	$\mathbf{b}_i$	The bias vector paired with its corresponding weight matrix $\mathbf{W}_i$
$\mathcal{A}^{(d)}_{feature}$	The output representation from the Feature-Mixing sub-layer	$\oplus$	The operation for element-wise addition
$\hat{\mathcal{A}}^{(d)}$	The final output tensor of the complete WTSMixer architecture	$*$	The operation for matrix multiplication
$\mathcal{A}^{(d)}_{mixing}$	The representation after the K-th (final) mixing block		

Appendix B. Hyperparameter Optimization

predict_length	sequence_length	num_blocks	dropout_rate	samples_per_batch	hidden_layer_dim	learning_rate	MAPE (%)	optimal_total_cost
H	6	1,2,3,4	0.1, 0.3, 0.5, 0.7, 0.9	2, 4	16, 32, 64, 128, 256	1e-5, 1e-4		
3	6	1	0.1	2	16	1e-5	79.33966	14,695,528.69
3	6	1	0.1	2	16	1e-4	79.33982	14,695,511.85
3	6	1	0.1	2	32	1e-5	79.3413	14,695,372.46
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
3	6	3	0.1	4	256	1e-4	27.5163	**304,763.09**
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
3	6	4	0.9	4	256	1e-5	144.58824	49,432,298.5
3	6	4	0.9	4	256	1e-4	143.51299	49,167,842.06

Table B.10: Results of Hyperparameter Optimization for WTSMixer Models under Scenario 1 (Best-performing values are indicated in **bold**).

predict_length	sequence_length	num_blocks	dropout_rate	samples_per_batch	hidden_layer_dim	learning_rate	MAPE (%)	optimal_total_cost
H	6	1,2,3,4	0.1, 0.3, 0.5, 0.7, 0.9	2, 4	16, 32, 64, 128, 256	1e-5, 1e-4		
3	6	1	0.1	1	16	1e-5	36.46066	714,872.09
3	6	1	0.1	1	16	1e-4	40.67522	559,163.33
3	6	1	0.1	1	32	1e-5	42.01536	775,199.47
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
3	6	3	0.1	4	256	1e-4	26.97368	**336,929.75**
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
3	6	4	0.9	4	256	1e-5	104.8307	967,151.14
3	6	4	0.9	4	256	1e-4	68.89641	708,537.81

Table B.11: Results of Hyperparameter Optimization for WTSMixer Models under Scenario 2 (Best-performing values are indicated in **bold**).

predict_length	sequence_length	num_blocks	dropout_rate	samples_per_batch	hidden_layer_dim	learning_rate	MAPE (%)	optimal_total_cost
H	6	1,2,3,4	0.1, 0.3, 0.5, 0.7, 0.9	2, 4	16, 32, 64, 128, 256	1e-5, 1e-4		
3	6	1	0.1	1	16	1e-5	35.08981	745,224.67
3	6	1	0.1	1	16	1e-4	38.91385	538,221.53
3	6	1	0.1	1	32	1e-5	45.26014	875,142.51
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
3	6	4	0.1	4	256	1e-4	28.69847	**339,145.41**
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
3	6	4	0.9	4	256	1e-5	104.81487	989,960.55
3	6	4	0.9	4	256	1e-4	67.91408	715,109.98

Table B.12: Results of Hyperparameter Optimization for WTSMixer Models under Scenario 3 (Best-performing values are indicated in **bold**).