# Multi-Horizon Forecasting of Inbound Inventory Volume (CBM) for IT Products Using Machine Learning

Arnonnut Yaothanee
*Faculty of Information Technology*
*Thai-Nichi Institute of Technology*
Bangkok, Thailand
2463110102@tni.ac.th

Napaphat Vichaidis
*Faculty of Information Technology*
*Thai-Nichi Institute of Technology*
Bangkok, Thailand
*Corresponding author :
napaphat@tni.ac.th

Kanda Tiwatthanont
*Faculty of Information Technology*
*Thai-Nichi Institute of Technology*
Bangkok, Thailand
kanda@tni.ac.th

***Abstract*—Accurate forecasting of inbound inventory dimension, expressed in cubic meters (CBM), is critical for warehouse space planning and operational stability in IT product distribution. While machine learning approaches have shown strong potential in time series forecasting, their application to volumetric CBM forecasting under limited monthly observation settings remains underexplored. This study develops a machine learning framework to predict monthly inbound CBM using historical operational records from 2020 to 2024, yielding approximately 60 monthly observations. Logarithmic transformation is applied prior to modeling to stabilize variance and mitigate scale distortion. A direct multi-horizon forecasting strategy is implemented, where independent models are trained for t+1, t+3, and t+6 month horizons under a strict chronological split (train: 2020–2022, validation: 2023, test: 2024) to eliminate information leakage. Six machine learning models are evaluated: ensemble tree-based methods (Random Forest and XGBoost) and recurrent neural architectures (RNN, LSTM, GRU, and LSTM with Attention). Four classical time-series baselines—Naïve, Seasonal Naïve, Holt-Winters, and SARIMA—are included as comparative references. Model performance is assessed using MAE, MSE, RMSE, and MAPE. The results show that the machine learning model XGBoost delivers the highest accuracy for short-term forecasting (MAPE = 23.31% at t+1), while Random Forest demonstrates the lowest average MAPE across all horizons (26.02%), reflecting greater stability at medium and longer lead times. Notably, SARIMA emerges as a competitive baseline, achieving the lowest average MAPE overall (23.78%), particularly excelling at t+3 (19.69%), which highlights the strength of well-fitted statistical models under limited data conditions. Ensemble tree-based methods outperform Naïve and Seasonal Naïve baselines across all horizons and show competitive performance against SARIMA at the t+1 horizon. Deep learning architectures exhibit higher performance variability under the constrained observation setting. These results suggest that model selection for monthly CBM forecasting under limited data should consider both forecast horizon and data availability, with ensemble methods offering consistent multi-horizon stability and SARIMA providing a strong statistical reference.***

***Keywords*—Inbound inventory dimension, cubic meter (CBM), time series forecasting, machine learning, direct multi-horizon forecasting, XGBoost, Random Forest, LSTM, Attention, warehouse planning, SARIMA**

## I. Introduction

Effective warehouse operations in IT product distribution require reliable anticipation of inbound inventory dimension. In this study, inbound inventory dimension is defined as the aggregated cubic measurement in cubic meters (CBM), representing the total storage space requirement of inbound products within a given month. Accurate forecasting of inbound CBM is critical for warehouse capacity planning, including space allocation and workforce preparation, as large forecast errors may lead to congestion risks or inefficient utilization of warehouse resources [1]. Inbound CBM often exhibits irregular fluctuations and time-varying volatility, which complicates predictive modeling [2]. Machine learning approaches have demonstrated strong potential in structured time series forecasting tasks, particularly when nonlinear patterns and feature-rich representations are present [3]. However, their effectiveness in volume-oriented logistics settings, such as inbound CBM forecasting, remains underexplored compared to demand quantity forecasting [2]. Most existing studies focus on unit-level demand or sales forecasting, while operational volume metrics such as CBM have received limited attention in the machine learning forecasting literature.

This study aggregates outsourced operational records from an IT product distribution company into a monthly time series spanning 2020 to 2024, yielding approximately 60 monthly observations. The monthly aggregation aligns the forecasting target with warehouse planning cycles and provides a consistent basis for multi-step forecasting, though it also limits the effective sample size available for model training, which may affect generalization of data-hungry deep learning architectures [4]. Machine learning models can learn nonlinear relationships from engineered temporal features, including lag variables and rolling statistics. In this work, tree-based ensemble methods, specifically Random Forest [5] and XGBoost [6], and recurrent neural network architectures, including RNN [8], LSTM [9], GRU [14], and LSTM with Attention [11], [12], are evaluated to reflect the two primary model families implemented in the experiments. Their performance is examined under different forecast horizons and a strict time-ordered data split to preserve temporal integrity [3]. In addition, the models are compared against classical time-series baseline methods to provide a comprehensive benchmark for performance evaluation.

Accordingly, this study evaluates multiple machine learning and deep learning models for forecasting the inbound inventory dimension of IT products. A direct multi-horizon forecasting design is applied, where separate models are trained for t+1, t+3, and t+6 month predictions [3]. The experiments follow a chronological split (train: 2020–2022, validation: 2023, test: 2024) to preserve temporal integrity and avoid information leakage. The primary contributions of this study are as follows:

(1) A structured machine learning framework for forecasting monthly inbound inventory dimension (CBM) in IT product distribution is proposed and empirically evaluated.

(2) A direct multi-horizon forecasting strategy across three planning horizons (t+1, t+3, t+6) is systematically compared across six model architectures spanning ensemble and recurrent approaches.

(3) Empirical findings under limited monthly observations reveal that well-fitted statistical models such as SARIMA remain competitive with ensemble tree-based methods, while both outperform deep learning architectures, providing practical guidance for model selection based on forecast horizon and data availability in warehouse planning contexts.

## II. Related Work

Forecasting in inventory-driven logistics systems must be evaluated not only for statistical accuracy but also for operational usability, since prediction errors may propagate into warehouse capacity allocation and workforce planning. This section reviews prior work across four areas relevant to this study: (1) general inventory and supply chain forecasting, (2) tree-based ensemble methods, (3) recurrent neural network architectures, and (4) operational forecasting applications.

### *A. Inventory and Supply Chain Forecasting*

Prior research emphasizes a persistent gap between inventory forecasting research and practical deployment, where model choices are constrained by volatility, limited history, and decision timelines [1]. Supply chain forecasting surveys highlight that data conditions and implementation context often shape the feasibility of advanced machine learning and deep learning methods in real operations [2]. Evidence from large-scale forecasting benchmarks indicates that no single model consistently dominates across datasets or horizons, reinforcing the importance of rigorous validation and horizon-aware evaluation [3]. Accordingly, forecast accuracy should be measured using multiple complementary metrics. Classic accuracy measures such as MAE, MSE, RMSE, and MAPE remain widely adopted; however, their interpretation depends on scale behavior and volatility characteristics, and thus they should be aligned with operational tolerance requirements [4]. Classical statistical methods—including Naïve, Seasonal Naïve, Holt-Winters exponential smoothing, and SARIMA—remain standard benchmarks in inventory and supply chain forecasting, as they are interpretable, computationally inexpensive, and provide a performance reference against which more complex models can be rigorously compared [4].

Intermittent demand forecasting remains a recognized challenge; neural forecasting methods and simplified neural architectures have been proposed to address irregular demand patterns, highlighting the role of model complexity and data availability in determining generalization [18], [19]. Notably, most existing inventory forecasting studies focus on unit-level demand or sales quantity, whereas volume-oriented metrics such as cubic meter (CBM) aggregation have received limited dedicated attention in the machine learning forecasting literature. This gap motivates the present study, which targets inbound CBM as the primary forecasting objective.

### *B. Tree-Based Ensemble Methods*

For structured time series forecasting formulated as supervised regression, tree-based ensemble learners provide strong baselines under engineered temporal features. Random Forest reduces variance via bootstrap aggregation and randomized feature selection, offering stability under noise and moderate sample sizes [5]. Gradient boosting frameworks such as XGBoost improve predictive accuracy through additive optimization with regularization, enabling effective learning of nonlinear interactions within lagged and calendar-based feature spaces [6]. Empirical evidence in applied sales prediction further suggests that machine learning approaches can outperform classical forecasting baselines when nonlinear patterns and feature-rich representations are present [7].

### *C. Recurrent Neural Network Architectures*

Sequential deep learning approaches model temporal dependency directly from ordered observations. Recurrent neural networks introduce feedback connections to encode temporal context [8], while Long Short-Term Memory (LSTM) improves learning stability through gated memory mechanisms that mitigate vanishing gradients and support longer-term dependency modeling [9]. Hybrid methods combining statistical smoothing with recurrent neural networks have demonstrated competitive performance in benchmark forecasting settings, suggesting that stabilizing level and trend components can improve robustness in practice [10].

Recent studies extend recurrent architectures using attention mechanisms to emphasize informative historical segments. Dual-stage attention-based recurrent models propose adaptive weighting strategies for time series prediction [11], and applied LSTM-attention variants have shown potential improvements in multivariate prediction scenarios [12]. In demand forecasting contexts, hyperparameter-optimized LSTM models have been reported to improve generalization under seasonal and irregular patterns [13], while comparative investigations indicate that GRU can achieve performance comparable to LSTM with reduced structural complexity in certain inventory-related settings [14].

### *D. Operational Forecasting Applications*

Operational forecasting applications across multiple domains reinforce the importance of evaluation design and stability. Hybrid ensembles have been employed for building load forecasting using real-time signals [15], and inventory-oriented demand forecasting has been developed for industrial settings such as the steel sector [16]. Retail forecasting research has also explored recurrent architectures alongside transformer-based models, indicating that architecture choice may impact stability depending on horizon characteristics and temporal structure [17].

Within this context, the present study evaluates a structured temporal feature set composed of calendar indicators and lag/rolling summaries for forecasting inbound inventory dimension (CBM) in IT product distribution. Both ensemble-based methods (Random Forest and XGBoost) and recurrent neural architectures (RNN, LSTM, GRU, and LSTM with Attention) are examined under a direct multi-horizon framework (t+1, t+3, t+6) to assess predictive stability across short- and medium-term planning intervals. Unlike prior studies that primarily target demand quantity, this work addresses volumetric CBM forecasting under a limited monthly observation setting, contributing empirical evidence on model suitability under constrained data conditions. TABLE *I* summarizes selected recent applied studies used as comparative context.

TABLE I SUMMARY OF SELECTED APPLIED FORECASTING STUDIES USING MACHINE LEARNING AND DEEP LEARNING METHODS

| No | Year | Objective | Technique | Dataset | Results |
|---|---|---|---|---|---|
| 1 | 1990 [8] | Sequential pattern learning | RNN (Elman network) | Artificial sequence data | Demonstrated temporal structure learning |
| 2 | 2017 [11] | Time series prediction | Dual-stage Attention RNN | Multiple financial & sensor datasets | Outperformed seq2seq baselines |
| 3 | 2020 [10] | Time series forecasting | Hybrid Exponential Smoothing + RNN | M4 Competition dataset | Ranked 1st in M4 Competition |
| 4 | 2020 [13] | Demand forecasting | Stacked LSTM + grid search | Furniture company monthly sales (132 months) | SMAPE = 0.1085 (proposed) vs 0.1122 (SVM) |
| 5 | 2021 [18] | Intermittent demand forecasting | Neural forecasting methods | Retail intermittent demand | Improved accuracy over classical methods |
| 6 | 2022 [7] | Sales forecasting | XGBoost vs ML + classical | Horticultural product sales | MAPE = 23.98% (XGBoost); 52.87% (classical) |
| 7 | 2022 [14] | Inventory demand forecasting | LSTM vs GRU comparative study | Industrial demand series | GRU competitive with LSTM at lower complexity |
| 8 | 2022 [17] | Grocery sales forecasting | Trimmed seq2seq + mod. Transformer | Corporación Favorita (day/store/item) | RMSLE = 0.538–0.542 |
| 9 | 2023 [12] | Multivariate time-series prediction | Enc–Dec LSTM + Attention | AIR / ELECTRICITY / STOCK / GOLD | RMSE = 0.013, MAPE = 0.76% (best) |
| 10 | 2024 [15] | Building load forecasting | Hybrid (LSTM, XGBoost, RF, LR) | Commercial & academic building load | RMSE = 2.99, MAE = 2.18 |
| 11 | 2024 [16] | Demand forecasting (inventory) | Ensemble (RNN–LSTM best) | Steel wire mesh demand series | RMSE = 45,931; MAPE = 1.20% |

## III. METHODOLOGY

This section describes the methodological framework adopted for forecasting inbound inventory dimension (CBM) of IT products. The approach integrates data engineering, temporal feature construction, supervised sequence modeling, and multi-horizon evaluation within a structured experimental design. Particular emphasis is placed on transforming outsourced operational records into a reliable and prediction-ready dataset, ensuring chronological integrity and minimizing information leakage. The overall framework consists of data preparation, feature engineering, forecasting formulation, model configuration, and performance evaluation.

### *A. Problem Definition*

Inbound inventory dimension, expressed in cubic meters (CBM), represents the aggregated storage capacity required for incoming IT products within a monthly planning period. Accurate forecasting of this metric supports warehouse capacity planning and resource allocation.

Let $\{y_1, y_2, \ldots, y_t\}$ denote the historical monthly inbound inventory dimension time series. The objective of this study is to estimate future values $y_{t+h}$ for forecasting horizons $h \in \{1, 3, 6\}$, corresponding to one-, three-, and six-month ahead predictions.

$$\hat{y}_t + h = f(x_t) \quad (1)$$

The forecasting problem is formulated as a supervised nonlinear regression task, where the model learns a functional mapping from the input representation at time t, denoted as $x_t$, to the predicted inbound inventory dimension at time t + h. This formulation explicitly reflects the multi-horizon prediction objective. A direct multi-horizon strategy is adopted, whereby independent models are constructed for each forecasting horizon. This approach prevents recursive error accumulation and ensures consistent evaluation across different lead times.

### *B. Data Set and Data Engineering*

The dataset used in this study was obtained from the operational logs of an IT product distributor in Thailand under a formal confidentiality agreement. Due to non-disclosure requirements, the specific identity of the organization and the brand name are strictly anonymized. The data comprises 60 monthly observations from 2020 to 2024, focusing on the inbound inventory dimension (CBM) of a representative line of IT hardware, such as printers and peripheral equipment. This ensures the study captures the logistics patterns and volumetric characteristics typical of the technology distribution sector.

To align with warehouse planning cycles, daily inbound CBM values are aggregated into monthly totals. Calendar and operational indicators are incorporated, including Month, Year, Holiday, WeekendFlag, and FirstMonth, where FirstMonth identifies the first working day of each month based on Holiday and WeekendFlag.

To improve numerical stability, the monthly CBM target is transformed using logarithmic scaling. No anomalous observations were removed from the series; the logarithmic transformation was deemed sufficient to stabilize the variance of irregular volume spikes, including those potentially attributable to supply chain disruptions during 2020. After temporal features are constructed, initial rows with missing values caused by lag and rolling operations are removed. The complete data engineering workflow, from raw daily records to the final model-ready dataset, is illustrated in Fig. 1.

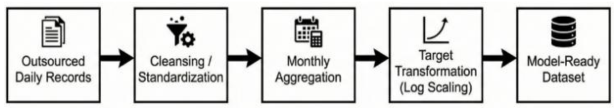


Fig. 1. Data engineering workflow for transforming outsourced inbound daily records into a monthly prediction-ready dataset

### *C. Feature Construction*

After constructing the monthly inbound inventory dimension series, temporal features are constructed to transform the forecasting task into a supervised learning problem . The feature representation integrates historical dependence and calendar information so that the same input structure can be applied across machine learning and deep learning models.

Historical dependence is captured using lag and rolling statistics derived from the monthly target. Lag_1, Lag_2, and Lag_3 represent short-term temporal memory, while Roll_3, Roll_6, and Roll_12 summarize recent and seasonal trends using moving averages from preceding months. For each forecasting horizon of 1, 3, and 6 months, features are constructed using data strictly up to the current month to provide a direct mapping to the target value. This ensures the input representation captures specific patterns relevant to each planning lead time . Calendar information is represented using Month and Year, while operational day-type signals include Holiday, WeekendFlag, and FirstMonth, where FirstMonth indicates the first working day of each month. Together, these variables form the input representation used for multi-horizon forecasting. The complete feature set is summarized in *Table II*.

TABLE II FEATURE SUMMARY

| Feature | Type | Formula / Definition | Purpose |
|---|---|---|---|
| Lag_1 | Lag | y(t−1) — log CBM, 1 month prior | Short-term memory |
| Lag_2 | Lag | y(t−2) — log CBM, 2 months prior | Short-term memory |
| Lag_3 | Lag | y(t−3) — log CBM, 3 months prior | Short-term memory |
| Roll_3 | Rolling | mean(y(t−3),…,y(t−1)) | Recent trend (3-month) |
| Roll_6 | Rolling | mean(y(t−6),…,y(t−1)) | Medium trend (6-month) |
| Roll_12 | Rolling | mean(y(t−12),…,y(t−1)) | Annual seasonal trend |
| Month | Calendar | Integer 1–12 | Seasonal pattern |
| Year | Calendar | Integer (2020–2024) | Long-term trend |
| Holiday | Operational | Binary: 1 if public holiday in that month | Demand disruption signal |
| Weekend Flag | Operational | Binary: 1 if weekend proportion exceeds working-day threshold | Working-day adjustment |
| FirstMonth | Operational | Binary: 1 if first working day of month (derived from Holiday and WeekendFlag) | Inbound scheduling signal |

For deep learning models (RNN, LSTM, GRU, and LSTM with Attention), all input features were additionally normalized using Min-Max scaling to the range [0, 1], with scaling parameters fitted exclusively on the training partition and applied identically to the validation and test sets.

### *D. Forecasting Design*

This study applies a direct multi-horizon forecasting design for horizons of 1, 3, and 6 months ahead. For each horizon, supervised samples are constructed using a fixed sequence length of 12 months. At any given time, the input consists of the feature representation from the previous 12 months, and the target is defined as the inbound inventory dimension at the forecast month. This design yields independent training targets for each horizon and avoids recursive propagation of prediction errors.

Model evaluation follows a strict time-ordered split to preserve chronological integrity. Monthly observations from 2020 to 2022 are used for training, 2023 is used for validation during hyperparameter tuning, and 2024 is reserved for final testing. To strictly prevent information leakage, all preprocessing parameters, including logarithmic and scaling factors, are derived solely from the training set. Furthermore, feature construction at each time point excludes any data from the forecast gap, ensuring no look-ahead bias is introduced during the multi-horizon modeling process. This time-based partition reflects realistic deployment conditions, where only past information is available at prediction time, and it minimizes the risk of information leakage.

### *E. Classical Baseline Methods*

To provide a comparative reference for the proposed machine learning framework, four classical time-series baseline methods are included in the evaluation. These methods represent standard benchmarks widely used in inventory and supply chain forecasting literature [4] and are fitted on the same training period (2020–2022) with parameters selected using the validation year (2023).

Naïve Forecast uses the most recent observed value as the prediction for all future horizons: $\hat{y}(t+h) = y(t)$. This represents the simplest possible forecast and serves as a lower bound for acceptable model performance.

Seasonal Naïve Forecast uses the observed value from the same month in the previous year: $\hat{y}(t+h) = y(t-12+h)$. This baseline captures annual seasonal patterns without parameter estimation.

Holt-Winters Exponential Smoothing (Triple Exponential Smoothing) models level, trend, and seasonal components using additive decomposition. Parameters ($\alpha$, $\beta$, $\gamma$) are optimized by minimizing the sum of squared errors on the training set. A seasonal period of 12 months is used, consistent with monthly aggregation [20].

SARIMA (Seasonal AutoRegressive Integrated Moving Average) is specified as SARIMA(1,1,1)(1,1,0)[12] based on preliminary ACF/PACF inspection of the training series. Model selection is confirmed by minimizing the Akaike Information Criterion (AIC) on the training partition [20].

All baseline forecasts are generated independently for each horizon $h \in \{1, 3, 6\}$ under the same chronological split protocol used for machine learning models, ensuring a fair comparison.

### *F. Model Development*

This study evaluates six forecasting models under a direct multi-horizon design (t+1, t+3, t+6), spanning feature-based machine learning and sequence-based deep learning approaches.

Random Forest (RF): Random Forest is an ensemble method built upon decision trees using bootstrapped sampling and feature randomization, which helps reduce overfitting and improves predictive stability for structured inputs [5].

Extreme Gradient Boosting (XGBoost): XGBoost is a gradient-boosted decision tree ensemble that sequentially trains multiple trees, where each new tree focuses on correcting the errors of prior learners, often yielding strong accuracy for nonlinear regression tasks [6].

Recurrent Neural Network (RNN) is a recurrent architecture designed for sequential data, where the hidden state propagates information over time to learn temporal dependence from historical patterns [8].

Long Short-Term Memory (LSTM): LSTM extends standard RNNs by incorporating gating mechanisms that support learning longer-range dependencies, which is beneficial for time series with trend or seasonal behavior [9]. LSTM with Attention: Attention-enhanced LSTM is included to examine whether selectively emphasizing informative temporal segments improves prediction accuracy. An attention layer is appended after the LSTM encoder to compute a weighted sum over hidden states, producing a context vector that guides the final output prediction. This formulation follows attention-based recurrent approaches proposed for time series prediction [11], [12].

Gated Recurrent Unit (GRU): GRU is a gated recurrent model that addresses vanishing/exploding gradient issues through gate-based control of information flow and is commonly compared with LSTM in forecasting applications [14].

For each forecasting horizon, hyperparameter tuning is conducted using grid search on the validation year (2023). The selected configuration is subsequently retrained on the combined training and validation period (2020–2023) and evaluated on the 2024 test set. TABLE *III* summarizes the search spaces and selected hyperparameters for the t+1, t+3, and t+6 horizons. The grid search produced identical configurations for t+1 and t+3 across all models, indicating consistent optimal settings under shorter forecast intervals. In contrast, several models selected different configurations at t+6, reflecting the increased complexity of longer-horizon forecasting.

TABLE III HYPERPARAMETER TUNING RESULTS SELECTED BY GRID SEARCH

| Model | Parameter | Search Space | t+1 | t+3 | t+6 |
|---|---|---|---|---|---|
| RNN | Hidden Units | {16, 32, 64} | 32 | 64 | 64 |
| | Dropout | {0.1, 0.2, 0.3} | 0.3 | 0.3 | 0.3 |
| | Sequence Length | 12 | 12 | 12 | 12 |
| LSTM | Hidden Units | {16, 32, 64} | 32 | 32 | 32 |
| | Dropout | {0.1, 0.2, 0.3} | 0.1 | 0.2 | 0.1 |
| | Sequence Length | 12 | 12 | 12 | 12 |
| LSTM with Attention | Hidden Units | {16, 32, 64} | 32 | 16 | 16 |
| | Dropout | {0.1, 0.2, 0.3} | 0.3 | 0.1 | 0.3 |
| | Attention Units | {16, 32} | 16 | 16 | 32 |
| | Sequence Length | 12 | 12 | 12 | 12 |
| GRU | Hidden Units | {16, 32, 64} | 16 | 64 | 64 |
| | Dropout | {0.1, 0.2, 0.3} | 0.2 | 0.3 | 0.1 |
| | Sequence Length | 12 | 12 | 12 | 12 |
| Random Forest | n_estimators | {100, 200, 300} | 100 | 100 | 300 |
| | max_depth | {3, 5, 7} | 7 | 3 | 5 |
| | min_samples_split | {2, 4} | 2 | 4 | 2 |
| XGBoost | n_estimators | {50, 100, 200, 300} | 200 | 200 | 100 |
| | max_depth | {2, 3, 5} | 2 | 2 | 5 |
| | subsample | {0.8, 1.0} | 0.8 | 1 | 0.8 |

## G. Evaluation Metrics

To evaluate forecasting accuracy for monthly inbound inventory volume (CBM), four error metrics are reported: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). Metrics are computed separately for each forecasting horizon (t+1, t+3, and t+6) to reflect horizon-dependent model behavior. Among these measures, MAPE is treated as the primary comparison metric because it expresses error in percentage terms, enabling clearer cross-model interpretation and operational communication [4].

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \quad (2)$$

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (3)$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \quad (4)$$

Mean Absolute Percentage Error (MAPE) expresses the average absolute error as a percentage of the actual value.

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \text{ x } 100 \quad (5)$$

## H. Experimental Environment

All experiments were conducted on a workstation equipped with an Intel Core i7 (12th Gen) CPU, 32 GB RAM, and an NVIDIA GeForce RTX 3060 Ti GPU. The models were implemented in Python 3.11 using standard scientific libraries, including NumPy and Pandas for data processing, scikit-learn for machine learning models, and TensorFlow/Keras for deep learning implementation.

# IV. EXPERIMENTAL RESULTS

## A. Overall ML Framework Performance

TABLE IV compares model performance on the 2024 test year under the year-based hold-out protocol across three forecasting horizons (t+1, t+3, t+6). Using MAPE as the primary criterion, ensemble models demonstrate strong and consistent performance. XGBoost achieves the best short-horizon accuracy at t+1 (MAPE = 23.31%), while Random Forest records the lowest errors at t+3 (MAPE = 24.07%) and t+6 (MAPE = 29.26%). The corresponding MAE and RMSE values further confirm the stability of these ensemble models as the forecasting horizon increases.

To ensure robustness despite the 60-observation limit, Early Stopping (patience = 10 epochs, monitored on validation loss) was applied to all recurrent architectures to prevent overfitting. Each deep learning model was independently trained five times using different random seeds, and the reported MAPE values represent the median across these runs to reduce initialization sensitivity. The higher variability observed in GRU (MAPE range: 27.54%–48.70% across horizons) and LSTM with Attention (27.98%–41.44%) confirms that recurrent architectures are susceptible to unstable convergence under limited training data. In contrast, tree-based ensemble methods (Random Forest and XGBoost) are deterministic given fixed hyperparameters and therefore exhibit consistent results across all horizons without requiring repeated runs.

TABLE IV TEST PERFORMANCE ACROSS FORECAST HORIZONS

| Model | Horizon | MAE | MSE | RMSE | MAPE (%) | Avg MAPE (%) |
|---|---|---|---|---|---|---|
| XGBoost | t+1 | 349.47 | 207,649 | 455.68 | 23.31 | 29.83 |
| | t+3 | 463.44 | 303,880 | 551.25 | 29.66 | |
| | t+6 | 624.19 | 517,956 | 719.78 | 36.51 | |

| Model | Horizon | MAE | MSE | RMSE | MAPE (%) | Avg MAPE (%) |
|---|---|---|---|---|---|---|
| Random Forest | t+1 | 368.75 | 215,484 | 464.20 | 24.72 | 26.02 |
| | t+3 | 387.21 | 237,511 | 487.35 | 24.07 | |
| | t+6 | 491.40 | 372,805 | 610.58 | 29.26 | |
| LSTM-Attention | t+1 | 418.61 | 269,206 | 518.86 | 27.98 | 32.69 |
| | t+3 | 431.92 | 271,133 | 520.75 | 28.66 | |
| | t+6 | 701.82 | 584,613 | 764.62 | 41.44 | |
| LSTM | t+1 | 462.37 | 294,719 | 542.88 | 31.18 | 33.15 |
| | t+3 | 522.06 | 356,979 | 597.48 | 34.54 | |
| | t+6 | 569.34 | 420,154 | 648.19 | 33.72 | |
| GRU | t+1 | 736.52 | 623,351 | 789.54 | 48.70 | 39.24 |
| | t+3 | 401.88 | 249,823 | 499.82 | 27.54 | |
| | t+6 | 706.47 | 595,823 | 771.88 | 41.49 | |
| RNN | t+1 | 771.33 | 667,889 | 817.29 | 51.18 | 46.15 |
| | t+3 | 633.11 | 449,985 | 670.73 | 41.71 | |
| | t+6 | 748.53 | 648,508 | 805.29 | 45.55 | |

TABLE V TEST MAPE (%) — CLASSICAL BASELINES VS. BEST ML MODELS

| Model | t+1 MAPE (%) | t+3 MAPE (%) | t+6 MAPE (%) | Avg MAPE (%) |
|---|---|---|---|---|
| Naïve | 46.08 | 34.82 | 44.32 | 41.74 |
| Seasonal Naïve | 26.69 | 41.00 | 45.92 | 37.87 |
| Holt-Winters | 25.44 | 25.44 | 25.44 | 25.44 |
| SARIMA | 23.37 | 19.69 | 28.28 | 23.78 |
| XGBoost (proposed) | 23.31 | 29.66 | 36.51 | 29.83 |
| Random Forest (proposed) | 24.72 | 24.07 | 29.26 | 26.02 |

The comparison between the proposed machine learning models and classical baselines is presented in Table V. The lowest average MAPE of 23.78% was achieved by SARIMA, with superior performance observed at t+3 (19.69%). Within the machine learning family, the best short-term accuracy at t+1 (23.31%) was yielded by XGBoost, while the most consistent performance across all horizons was maintained by Random Forest (Avg MAPE = 26.02%). Significant predictive improvements over Naive and Seasonal Naive strategies were provided by all proposed models. Furthermore, the stability of Holt-Winters (Avg MAPE = 25.44%) and SARIMA suggests that classical statistical methods remain highly effective in capturing patterns within data-constrained environments, serving as strong contenders alongside machine learning frameworks.

*B. Comparison with the actual results*

Fig. 2 illustrates the monthly actual and predicted inbound CBM values for the 2024 test year under the t+1 horizon using XGBoost. The model effectively follows the overall seasonal direction and captures major turning points throughout the year. While moderate underestimation is observed during mid-year peak fluctuations, the predicted curve remains closely aligned with the actual series in most months, supporting the quantitative findings reported in Table IV.

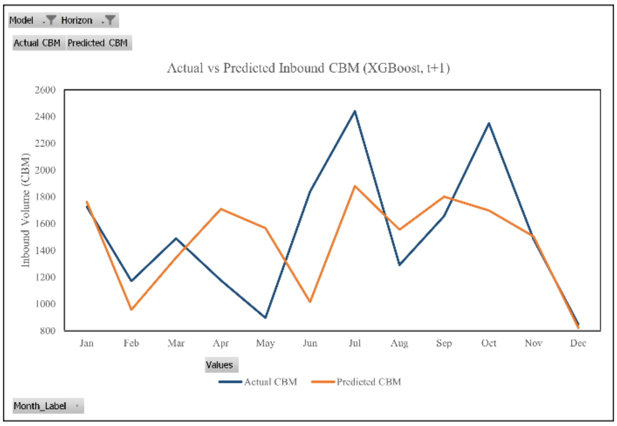


Fig. 2 Actual versus predicted inbound CBM for the 2024 test year using XGBoost under the t+1 horizon.

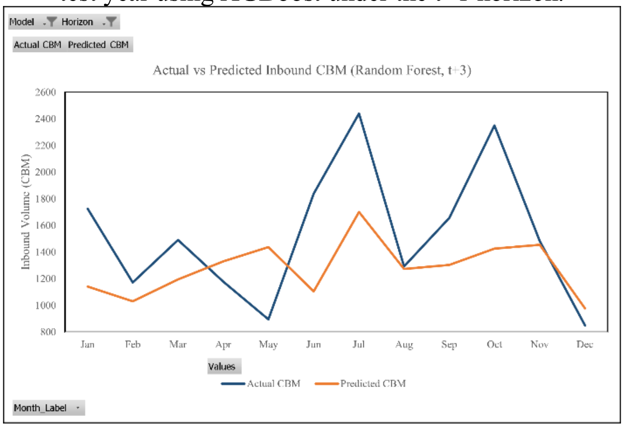


Fig. 3. Actual versus predicted inbound CBM for the 2024 test year using Random Forest under the t+3 horizon.

Fig. 3 presents the actual versus predicted monthly inbound CBM for the 2024 test year using Random Forest under the t+3 forecast horizon. The model successfully captures the overall seasonal direction across the majority of test months, maintaining a MAPE of 24.07% — the lowest among all models at this horizon.

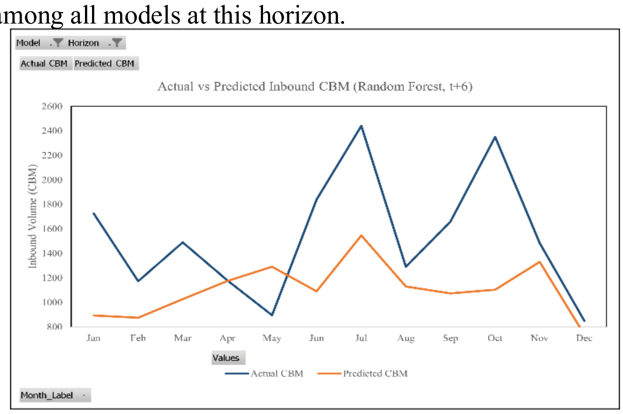


Fig. 4. Actual versus predicted inbound CBM for the 2024 test year using Random Forest under the t+6 horizon.

Fig. 4 illustrates the actual versus predicted inbound CBM for the 2024 test year using Random Forest under the t+6 forecast horizon. The model maintains general directional alignment but exhibits wider deviations during peak-demand months, reflected in a MAPE of 29.26% — the lowest among all evaluated models at this horizon.

### C. Ensemble vs. Deep Learning under Limited Data

A cross-horizon comparison highlights the horizon-dependent nature of model performance. Among all evaluated models, SARIMA achieves the lowest average MAPE (23.78%) and dominates at t+3 (19.69%), indicating strong suitability for medium-term forecasting when seasonal patterns are regular. Within the ML family, Random Forest achieves the lowest average MAPE (26.02%), followed by XGBoost (29.83%), indicating stronger stability across multiple horizons compared to deep learning models. XGBoost matches SARIMA closely at t+1 (23.31% vs. 23.37%), making it the preferred ML choice for short-term scheduling. Among deep learning models, LSTM with Attention records the best performance within its group (32.69%) and performs particularly well at t+3 (28.66%), although it remains above SARIMA and Random Forest at that horizon. GRU shows inconsistent behavior, performing poorly at t+1 but achieving competitive accuracy at t+3. Overall, the results suggest that under limited monthly observations, classical statistical methods — particularly SARIMA — remain strong contenders, while tree-based ensemble models provide more consistent ML-based forecasting than deep learning architectures.

## V. Discussion

### A. Interpretation of Key Findings

The results reveal a nuanced performance pattern across model families. Tree-based ensemble methods consistently outperform recurrent neural architectures; however, a notable finding is that SARIMA achieves the lowest average MAPE (23.78%) and the best performance at t+3 (19.69%), surpassing all ML models at that horizon. This supports prior evidence that well-specified statistical models can perform competitively under limited data and stable seasonality [4].

With only 36 training observations (2020–2022), the sequence length is insufficient for stable convergence in recurrent models, leading to higher variability in GRU (MAPE: 27.54%–48.70%) and LSTM with Attention (27.98%–41.44%). Although this limitation also constrains tree-based methods, XGBoost remains competitive at t+1 (MAPE = 23.31%), while Random Forest provides the most stable performance across horizons. Additionally, identical optimal hyperparameters for t+1 and t+3, but not for t+6, suggest that short- and medium-term forecasts share similar temporal structures, whereas longer horizons require different representations.

### B. Comparison with Related Literature

The XGBoost result at t+1 (MAPE = 23.31%) is directly comparable to Haselbeck et al. [7], who reported MAPE = 23.98% for XGBoost applied to horticultural sales forecasting, and substantially lower than the classical baselines in the same study (MAPE = 52.87%). However, the present study reveals an important contrast: SARIMA achieves a lower average MAPE (23.78%) than both XGBoost (29.83%) and Random Forest (26.02%) when fitted on the same limited dataset. This divergence from Haselbeck et al. [7] likely reflects the difference in dataset size and the regularity of seasonal patterns in the monthly CBM series, which favors SARIMA's additive seasonal decomposition. This finding aligns with evidence from Makridakis et al. [3] that no single model consistently dominates across all settings, and that classical methods remain strong competitors under short, seasonal series. Random Forest's average MAPE of 26.02% is nonetheless consistent with ensemble approaches reported in building load and demand forecasting applications [15, 16], confirming its multi-horizon stability within the ML family.

Deep learning results in this study show markedly higher variability than those reported in LSTM-based demand forecasting benchmarks [13, 14]. Abbasimehr et al. [13] achieved SMAPE = 0.1085 on a furniture sales dataset with 132 monthly observations — nearly four times the training size available here. Li et al. [14] similarly worked with longer industrial demand series. This contrast directly supports the conclusion that data volume is the primary limiting factor for deep learning in this domain, rather than architectural suitability per se.

### C. Practical Implications for Warehouse Planning

The forecasting results have clear operational implications. At t+1, SARIMA (MAPE = 23.37%) and XGBoost (23.31%) provide reliable short-term forecasts. At t+3, SARIMA performs best (19.69%), supporting effective medium-term planning. At t+6, Random Forest (29.26%) offers stable ML performance, while Holt-Winters (25.44%) provides a strong statistical alternative.

A practical deployment strategy is therefore: (1) use SARIMA or XGBoost for short-term forecasting, (2) use SARIMA for medium-term planning, and (3) use Random Forest or Holt-Winters for long-term capacity planning. A safety buffer of 25–30% is recommended to manage forecast uncertainty [1].

### D. Limitations

Several limitations in this study should be recognized. First, the data was collected from a single IT distributor in Thailand, which means the results might not represent other companies or different types of products in other regions. Second, because the data was grouped by month, only 36 training points were available. This small sample size made it difficult for deep learning models to reach their best performance and prevented the analysis of daily or weekly patterns. Third, external factors such as economic indicators, promotion calendars, and shipping delays were not included in the models, which could affect the accuracy during unusual demand periods. Finally, although the models were run five times to check for stability, a full rolling-origin cross-validation was not performed because the testing period was limited to only 12 months.

## VI. Conclusion

This study successfully addressed all three research objectives through a systematic empirical evaluation of machine learning and deep learning models for multi-horizon forecasting of monthly inbound inventory dimension (CBM) in IT product distribution. A direct forecasting framework incorporating temporal feature engineering was applied across six model architectures — Random Forest, XGBoost, RNN, LSTM, GRU, and LSTM with Attention — over three planning horizons (t+1, t+3, t+6) using a strict chronological hold-out protocol (train: 2020–2022, validation: 2023, test: 2024).

The empirical results reveal that SARIMA achieves the lowest average MAPE overall (23.78%), with particularly strong performance at t+3 (19.69%), establishing it as the most accurate model in this study under the limited data setting. XGBoost achieves the best short-term ML accuracy (MAPE = 23.31% at t+1), closely matching SARIMA at that horizon, while Random Forest produces the most consistent ML performance across all horizons (Avg MAPE = 26.02%). Both ensemble models outperform Naïve, Seasonal Naïve, and all deep learning architectures evaluated, but do not universally surpass SARIMA and Holt-Winters. These MAPE values, ranging from approximately 23% to 30% for the best-performing models, correspond to operationally manageable uncertainty margins that support practical warehouse space reservation and capacity planning decisions.

Theoretical Contributions: Empirical evidence is provided that classical statistical models, specifically SARIMA, remain highly competitive against tree-based ensembles in data-constrained monthly forecasting environments. These findings refine the understanding of model suitability boundaries and challenge the assumption of universal machine learning dominance, consistent with large-scale benchmarks in [3]. Furthermore, the efficacy of the direct multi-horizon framework for volumetric CBM forecasting is demonstrated, extending its applicability beyond the unit-level demand contexts prevalent in existing literature [3, 7, 14].

Practical Contributions: Actionable model selection guidance is established for warehouse practitioners based on specific planning horizons. SARIMA is recommended for medium-term planning (t+3) due to its superior accuracy, while XGBoost is identified as optimal for short-term scheduling (t+1). For consistent performance across all horizons, Random Forest is preferred. The framework is designed for deployment with limited historical data, requiring only 36 months of training observations, thereby making advanced forecasting accessible to small and medium-sized distribution operations without extensive data infrastructure.

## Acknowledgment

The author would like to express sincere appreciation to the academic advisor for valuable guidance and constructive recommendations throughout the research process. The author also gratefully acknowledges the internal stakeholders and professionals who provided access to operational inbound data and supported data preparation for this study. Their cooperation and assistance were essential to the successful completion of this research.

## References

[1] T. E. Goltsos, A. A. Syntetos, C. H. Glock, and G. Ioannou, "Inventory–forecasting: Mind the gap," *European Journal of Operational Research*, vol. 299, no. 2, pp. 397–419, 2022, doi: 10.1016/j.ejor.2021.06.057.

[2] M. Seyedan and F. Mafakheri, "Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities," *Journal of Big Data*, vol. 7, art. 53, 2020, doi: 10.1186/s40537-020-00329-2.

[3] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 Competition: Results, findings, conclusion and way forward," *International Journal of Forecasting*, vol. 34, no. 4, pp. 802–808, 2018, doi: 10.1016/j.ijforecast.2018.06.001.

[4] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006, doi: 10.1016/j.ijforecast.2006.03.001.

[5] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[6] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, New York, NY, USA, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[7] F. Haselbeck, J. Killinger, K. Menrad, T. Hannus, and D. G. Grimm, "Machine learning outperforms classical forecasting on horticultural sales predictions," *Machine Learning with Applications*, vol. 7, p. 100239, 2022, doi: 10.1016/j.mlwa.2021.100239.

[8] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990, doi: 10.1207/s15516709cog1402_1.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[10] S. Smyl, "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting," *International Journal of Forecasting*, vol. 36, no. 1, pp. 75–85, 2020, doi: 10.1016/j.ijforecast.2019.03.017.

[11] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in *Proc. 26th Int. Joint Conf. Artificial Intelligence (IJCAI)*, Melbourne, Australia, 2017, pp. 2627–2633, doi: 10.24963/ijcai.2017/366.

[12] X. Wen and W. Li, "Time series prediction based on LSTM-attention-LSTM model," *IEEE Access*, vol. 11, pp. 48322–48331, 2023, doi: 10.1109/ACCESS.2023.3276628.

[13] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using LSTM network for demand forecasting," *Computers & Industrial Engineering*, vol. 143, p. 106435, 2020, doi: 10.1016/j.cie.2020.106435.

[14] Y. Li, Z. Zhang, and Y. Huang, "Deep learning for inventory demand forecasting: A comparative study using LSTM and GRU," *Computers & Industrial Engineering*, vol. 167, p. 108037, 2022, doi: 10.1016/j.cie.2022.108037.

[15] I. Rafiq, A. Mahmood, U. Ahmed, I. Aziz, A. R. Khan, and S. Razzaq, "A novel approach for forecasting and scheduling building load through real-time occupant count data," *Arabian Journal for Science and Engineering*, vol. 49, pp. 11507–11523, 2024, doi: 10.1007/s13369-024-09296-9.

[16] N. Sukolkit, S. Arunyanart, and A. Apichottanakul, "An open innovative inventory management based demand forecasting approach for the steel industry," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 10, p. 100407, 2024, doi: 10.1016/j.joitmc.2024.100407.

[17] I. Vallés-Pérez, E. Soria-Olivas, M. Martínez-Sober, A. J. Serrano-López, J. Gómez-Sanchís, and F. Mateo, "Approaching sales forecasting using recurrent neural networks and transformers," *Expert Systems with Applications*, vol. 201, p. 116993, 2022, doi: 10.1016/j.eswa.2022.116993.

[18] X. Tian, H. Wang, and E. Erjiang, "Forecasting intermittent demand for inventory management by retailers: A new approach," *Journal of Retailing and Consumer Services*, vol. 62, p. 102662, 2021, doi: 10.1016/j.jretconser.2021.102662.

[19] F. Lolli, R. Gamberini, A. Regattieri, E. Balugani, T. Gatos, and S. Gucci, "Single-hidden layer neural networks for forecasting intermittent demand," *International Journal of Production Economics*, vol. 183, pp. 116–128, 2017, doi: 10.1016/j.ijpe.2016.10.021.

[20] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed. Melbourne, Australia: OTexts, 2021. [Online]. Available: https://otexts.com/fpp3