AIMS Mathematics, 11(7): 21885–21927.
DOI: 10.3934/math.2026885
Received: 27 April 2026
Revised: 30 June 2026
Accepted: 13 July 2026
Published: 23 July 2026

http://www.aimspress.com/journal/Math

Research article

Covariate-aware transformer architectures for long-horizon demand forecasting in Retail analytics

Masad A. Alrasheedi[1], **Asamh Saleh M. Al Luhayb**[2,*] **and Nasser Aedh Alreshidi**[3]

[1] Department of Management Information Systems, College of Business Administration, Taibah University, Madinah, Saudi Arabia
[2] Department of Mathematics, College of Science, Qassim University, P.O. Box 6644, Buraydah, 51452, Saudi Arabia
[3] Department of Mathematics, College of Science, Northern Border University, Arar 73213, Saudi Arabia

* **Correspondence:** Email: a.alluhayb@qu.edu.sa.

Abstract: Demand forecasting from a retail perspective affects various facets of the business, including workforce planning, inventory control, and supply chain management, and informs key business strategies. The long-range demand forecasting techniques evaluate various external influences and deal with time series data that often exhibit irregular patterns and complex seasonality, and are often hierarchical. In this work, we proposed a long-range demand forecasting framework that used covariates and was based on transformer architectures with the following key components: multihead self-attention, direct multi-horizon prediction, positional encodings, and categorical embeddings. This framework can model various temporal patterns and was context-aware within the retail system. Having implemented the framework, we benchmarked it against state-of-the-art demand forecasting models on several retail forecasting datasets, as well as other complex time-series, multi-dimensional forecasting datasets. The model empirically demonstrated a 3.1% improvement on the retail forecasting benchmark (achieving a weighted root mean squared scaled error (WRMSSE) of 0.498) and a 1.7% improvement on the benchmark for complex time series datasets (achieving an symmetric mean absolute percentage error (sMAPE) of 12.47) compared to the leading transformer models. Additionally, we provided empirical evidence of enhanced demand forecasting, improved stability and robustness, reduced training time, and improved model throughput. With all of these model features and enhancements combined, we believe this will make retail demand forecasting significantly more feasible and add considerable operational value, particularly for the planning and strategy aspects of retail management.

Keywords: long-horizon forecasting; retail demand prediction; transformer models; time-series analytics; hierarchical forecasting; deep learning

Mathematics Subject Classification: 68T07, 68T09, 62M10, 90B50, 90C90

1. Introduction

In retail analytics, accurate long-term demand forecasting remains a primary challenge due to intricate seasonal dynamics and the complexities introduced by promotions and cross-item interactions within large hierarchical contexts. With the availability of detailed sales data, there has been a strong push to develop scalable models that can effectively forecast demand with nonlinear temporal dependencies over extended forecast horizons. There have also been several recent studies indicating that traditional statistical methods are less robust when using long forecast windows and large retail hierarchies [1].

Deep learning methods are growing in popularity to help solve these problems. In particular, newer neural forecasting models are more flexible in handling multiple inputs simultaneously and more efficient for large retail businesses [2]. On the other hand, long temporal contexts can cause recurrent neural networks to perform worse when processing data sequentially, leading to unstable gradients [3].

Newly developed transformer-based models are considered strong alternatives for time-series forecasting. The self-attention mechanism in these models can model direct dependencies over very long time periods without first computing them recursively. In 2024, empirical evaluations showed that attention-based models outperformed recurrent baseline methods on multi-horizon forecasting tasks [4]. Also, recent decomposition-enhanced transformers have demonstrated improved stability in modeling seasonal retail demand [5].

Transformer-based models show consistent improvements across a number of standardized benchmark forecasting datasets and at various forecasting frequencies and horizons [6]. Retail-oriented datasets whose sales patterns are sporadic, exhibit intermittent sales behavior, and, consequently, result in sparse demand demonstrate particularly large improvements over more traditional forecasting methods [7].

While various improvements have been made, ongoing issues still require resolution. Current transformer architectures struggle to scale to large, hierarchical retail structures. Similarly, determining how best to incorporate covariates, including price signals and calendar events, into transformer architectures is an ongoing area of research [8]. Finally, establishing long-term predictive accuracy with transformers will require comparative analyses to understand how performance typically decreases as the prediction horizon exceeds the length of the training window [9].

While architectures using transformers such as Informer, Autoformer, and PatchTST provide generalized long-range sequence modeling with competitive forecasting capabilities, many methodological gaps remain for large-scale forecasting of retail demand. Existing transformer-based forecasting architectures mostly focus on homogeneous temporal frameworks and lack mechanisms to combine modeling of retail hierarchies, covariate interactions, and stable long-range forecasting with erratic demand distributions. In addition, many studies focus on forecasting with different pre-processing pipelines, optimization, and computing frameworks. This makes it very difficult to discern the effects of certain architectures from implementation differences.

The proposed model provides a cohesive covariate-aware transformer structure for hierarchical

retail forecasting. This structure incorporates several innovations, such as multihead global temporal attention, categorical hierarchy-aware embeddings, temporal covariate fusion, positional sequence encoding, and multi-horizon direct optimization. This framework integrates these components into a single globally trained forecasting structure. Rather than using traditional autoregressive methods that iteratively propagate prediction errors over long forecast horizons, this architecture uses direct horizon-level forecasting, which promotes stability in long-term forecasts and minimizes cumulative uncertainty. This framework uses a unified representation-learning structure that combines temporal, inter-series, and covariate-based theoretical approaches for retail demand series. This structure allows the transformer encoder to embed inter-series learning alongside inter-series and contextual learning within a learning framework. For a fair comparison with all other forecasting architectures, all other architectures are built using the same preprocessing and optimization, and all structures are built on the same hardware and configured similarly.

The main goal of my research is to create an integrated transformer-based forecasting system to address real-world challenges in large-scale retail demand forecasting, such as diverse demand behavior, an imbalanced product hierarchy, long-range forecasting, and multiple contextual covariates. Most current forecasting systems focus on improving individual components of the transformer. However, my research will examine the extent of improvements in forecasting accuracy, robustness, and scalability, and, in particular, the cost-benefit of computational efficiency when temporal self-attention, hierarchical categorical representations, covariate-driven feature construction, and direct multi-horizon optimization are combined in a single system for real-world retail demand forecasting.

The main contributions of this study include:

- Development of a framework for hierarchical long-horizon retail demand forecasting utilizing covariate-aware transformers.
- Creation of a global representation learning approach.
- Design of a direct multi-horizon forecasting approach.
- Establishment of a benchmarking protocol and methods for experimental comparison.
- Illustration of the framework and approach across forecasting use cases, demonstrating improvement of convergence and forecasting robustness and stability.

The remainder of this paper is organized as follows. Section 1 introduces the problem background and motivation. Section 2 reviews related work in statistical, recurrent, and transformer-based time-series forecasting. Section 3 presents the proposed transformer framework. This includes problem formulation and architectural design. Section 4 describes the experimental setup and reports comprehensive results. These cover quantitative measures, ablation, and scalability. Section 5 concludes the paper and outlines future research directions.

2. Literature review

Long-term time-series forecasting has been a subject of great interest in recent years because it is essential for retail analytics, supply chain management, and strategic planning. Classical statistical methods such as exponential smoothing (ETS) and autoregressive integrated moving average (ARIMA) have historically been the dominant tools for demand forecasting because they are very interpretable and have proven to perform well on seasonal time series. These statistical models continue to compete

at the large-scale retail benchmarks such as the M4 and M5 competitions [10, 11]. However, they do not perform well when modeling complex nonlinear dependencies and interactions between different time series.

The rapidly evolving landscape of explainable artificial intelligence (AI), transformer-based representation learning, and intelligent decision-support systems underscores the need for scalable, interpretable deep learning systems. Earlier work relied on explainability-driven convolutional neural networks and gradient-weighted class activation mapping (XGrad-CAM) to enhance the interpretability and feature attribution of brain tumor classification systems and disease recognition systems [12, 13]. Next, attention-based hybrid convolutional neural network (CNN)–Transformer systems provided explainable attention and flexible feature interaction for improved representation in medical imaging and skin cancer diagnosis [14, 15]. Several multimodal systems based on transformers have successfully integrated cross-adaptive attention combined with hierarchical feature fusion to model different data forms [16]. Beyond the health sector, explainable intelligent decision-support systems and large-scale AI-based analytical systems highlight the essential role of adaptable, robust representation learning across diverse real-world conditions [17, 18]. These systems represent the shift toward integrated deep learning systems capable of modeling heterogeneous dependencies and enabling long-range, contextual, and interpretable decision-making. This work applies representation learning and attention-based modeling to long-horizon retail demand forecasting and offers a transformer framework with covariate awareness that fuses covariates with hierarchical representations, combines global self-attention, and supports scalable, direct forecasting across multiple successive horizons.

Building upon the strengths and limitations of classical models, deep learning models have progressively replaced purely statistical approaches in large-scale forecasting tasks. Recurrent neural networks (RNNs), particularly Long short-term memory (LSTM) networks, demonstrated improved performance in modeling sequential dependencies across retail demand series [19]. Subsequently, feed-forward architectures such as Neural basis expansion analysis for interpretable time series forecasting (N-BEATS) introduced deep residual stacks specifically designed for time-series forecasting and achieved competitive results on the M4 benchmark [20]. More recently, hierarchical interpolation-based architectures such as neural hierarchical interpolation for time series forecasting (N-HiTS) have further improved multi-horizon forecasting performance, especially in large-scale retail settings [21].

Despite these advances, recurrent and feed-forward architectures struggle with long-range temporal dependencies as sequence length increases. Transformer-based models were originally created for sequence modeling in natural language processing [22]. More recently, they have been adapted for long-horizon time-series forecasting. Informer introduced a sparse attention mechanism that reduces computational complexity for long sequences [23]. Autoformer added seasonal–trend decomposition directly within the transformer architecture, which improved long-term forecasting stability [24]. PatchTST proposed a patch-based tokenization strategy that enhances temporal representation learning and achieved strong results on large forecasting benchmarks [25].

Recent forecasting studies have shown that transformer-based architectures can improve long-sequence and multi-horizon prediction by modeling long-range temporal dependencies, decomposition patterns, patched temporal representations, and static or time-varying covariates [23–26]. These works provide an important foundation for the proposed covariate-aware transformer framework,

which further evaluates long-horizon retail forecasting under controlled benchmarking, robustness, scalability, and statistical validation settings.

Table 1. Comparative analysis of recent publications related to transformer-based and covariate-aware forecasting.

Study	Main Focus	Key Methodological Feature	Strength	Limitation
Transformer-based probabilistic forecasting [27]	Retail time-series forecasting with explanatory variables	Transformer models with probabilistic prediction and external covariates	Demonstrates the value of explanatory variables for retail forecasting	Mainly emphasizes probabilistic forecasting and does not provide extensive hierarchical benchmark validation
Covariate-informed transformer [28]	General covariate-aware time-series forecasting	Explicit modeling of covariates available across the forecasting horizon	Improves contextual representation of future-known variables	Primarily targets general forecasting and does not focus on large-scale retail hierarchy
Unified covariate adaptation [29]	Covariate adaptation for time-series foundation models	Homogenization and attention-based fusion of heterogeneous covariates	Provides a flexible strategy for adapting foundation models to covariate-rich settings	Requires adaptation to pretrained forecasting models and may involve higher integration complexity
Temporal fusion transformer for weekly retail sales [30]	Multi-horizon retail sales forecasting	Static store identifiers and time-varying exogenous variables with probabilistic outputs	Provides interpretable multi-horizon retail forecasting with external variables	Focuses on a single retail setting and limited forecasting frequency
Hybrid time-aware attention model [31]	E-commerce demand forecasting	Temporal convolutions, recurrent modeling, and self-attention	Combines local temporal patterns with attention-based long-range modeling	Uses a hybrid design rather than a unified transformer-based architecture
Proposed framework	Long-horizon hierarchical and heterogeneous demand forecasting	multihead self-attention, positional encoding, categorical embeddings, covariate integration, and direct multi-horizon prediction	Provides controlled benchmarking, horizon-wise analysis, ablation, robustness testing, statistical validation, and computational efficiency evaluation	Future work may extend the framework toward probabilistic uncertainty estimation and foundation-model adaptation

Recent publications have shed new light on long-horizon forecasting with the emphasis on transformer-based and covariate-aware learning. In the context of transformer-based probabilistic forecasting, the addition of context in the form of promotional, pricing, holiday, and socio-economic variables improves the retail demand forecasting beyond historical sales data [27]. Likewise, to help close the gap between known future covariates and historical target variables, contextual signals along the forecasting horizon are modeled within covariate-aware transformer frameworks [28]. In the

context of unified covariate adaptation, the attention-based integration of diverse covariates is essential in foundational forecasting models [29]. In the context of retail forecasting, existing studies indicate that using transformer architectures improves multi-horizon sales forecasting models when both static identifiers and time-varying exogenous variables are incorporated [30]. Nevertheless, most studies are limited to either probabilistic forecasting or single-domain retail covariates and/or general covariate adaptation, with no extended empirical validation that covers hierarchical retail and diverse seasonal benchmarks. In order to compare with the recent literature, Table 1 includes comparative studies of the recent publications in this area.

Recent empirical research has compared the performance of several methods for time series modeling (including classical statistical models, recurrent networks, and transformer-based architectures) using standard benchmarking datasets: M4 and M5 [32, 33]. These evaluations consistently show that transformer-based architectures outperform traditional statistical and recurrent modeling methods, particularly for long-horizon forecasting. This is demonstrated in Table 2, where transformer-based models have lower symmetric mean absolute percentage error (sMAPE) scores on M4 and better weighted root mean squared scaled error (WRMSSE) performance on M5 than traditional approaches.

Despite improvements in time-series forecasting enabled by advances in deep learning, significant gaps remain in long-term demand forecasting and retail analytics. First, the great majority of current models have been optimized for short- and medium-term forecasting and have been shown to exhibit instability in their error levels as forecast length increases. Second, many prior research studies have used different preprocessing, training, and computational environments, making their results less generalizable and reliable. Third, hierarchical retail databases introduce dependencies between series as well as variability within the series being forecast, which traditional recurrent networks or shallow feedforward architectures are unable to address effectively. Additionally, the robustness of forecasting models tested under distributional shifts, when transferred across data, and under noisy observations has not been systematically studied, despite their prevalence in the retail industry. In summary, these three limitations have created methodological ambiguity and limited the ability to scale a large-scale forecasting system.

Recent forecasting studies have sought to advance transformer-based and deep learning predictive models by incorporating decomposition-based attention, patch-based temporal representation, frequency-domain modeling, and linear forecasting frameworks. Proponents of these models suggest they allow for a greater range of forecasting horizons, enabling models to better account for long-range dependencies and seasonal and temporal patterns of varying type and structure. Retail demand forecasting models, however, encounter several challenges, particularly in joint modeling of temporal covariates, handling a multi-category hierarchy, accounting for varying levels of aggregation, and ensuring long-horizon stability. In light of these challenges, this research aims to develop a unified, covariate-aware transformer framework to address the issues outlined above in large-scale retail demand forecasting.

Table 2. Representative baseline forecasting models on M4 and M5 benchmarks. Lower values indicate better performance.

Method	Score	Key Limitations
M4 Benchmark (sMAPE)		
Statistical Benchmark (ETS/ARIMA) – Classical exponential smoothing and autoregressive seasonal models [10]	13.92	Limited nonlinear modeling; weak long-range dependency capture
LSTM – Recurrent neural network with gated memory units for sequential temporal modeling [19]	13.36	Sequential computation limits scalability; performance degrades for very long horizons
N-BEATS – Deep residual fully connected architecture with backward and forward residual stacks [20]	13.18	No explicit temporal attention; limited covariate integration
Informer – Sparse self-attention transformer using ProbSparse mechanism for long-sequence forecasting [23]	13.14	Attention approximation may reduce fine-grained dependency modeling
Autoformer – Decomposition-based transformer with auto-correlation attention for seasonal-trend separation [24]	12.97	Increased architectural complexity; sensitive to decomposition quality
PatchTST – Patch-based transformer encoding temporal segments as tokens for enhanced representation learning [25]	12.62	High memory usage for large-scale multivariate datasets
M5 Benchmark (WRMSSE)		
Statistical Baseline – Hierarchical ETS-based ensemble optimized for retail aggregation levels [11]	0.577	Struggles with nonlinear demand shifts and promotional dynamics
Prophet – Additive regression model combining trend and seasonal components [34]	0.565	Assumes additive structure; limited cross-series learning
LSTM – Sequence-to-sequence recurrent forecasting model [19]	0.548	Inefficient for long input windows; gradient instability in extended forecasts
DeepAR – Probabilistic autoregressive RNN with likelihood-based training [35]	0.542	Error accumulation during autoregressive rollout
N-HiTS – Hierarchical interpolation deep network for multi-horizon prediction [21]	0.531	Requires careful hyperparameter tuning; limited explicit long-range attention
Informer – Efficient transformer with sparse attention for large-scale forecasting [23]	0.524	Approximate attention may miss subtle demand variations
Autoformer – Seasonal-trend decomposition transformer architecture [24]	0.518	Increased computational overhead for large hierarchical retail systems

The proposed transformer-based approach closes these gaps via a comprehensive global learning framework tailored for long-term forecasts. Long-term temporal dependencies can be modeled via multiheaded self-attention without compounding errors through recursive time-series analyses, while

positional encoding preserves chronological order across long input time-frames. By using per-series normalization and categorical embeddings, variation in relative scale is reduced, and relationships between series can be identified in a hierarchical retail environment. Additionally, a direct multi-horizon forecast reduces estimation instability associated with stepwise autoregressive models. Finally, all models are developed within a consistent research context, enabling complete and unbiased evaluation across all experiments. Together, through this architecture, the approach proposes an integrated framework to support long-term retail demand forecasting by closing existing gaps in performance, evaluation, and scalability.

The uniqueness of our framework lies not in adding a single modular change to the architecture, but in its ability to provide a generalized forecasting system that encapsulates many complementary constituents and can be optimized in a modular way. From a methodological perspective, the architecture can jointly capture temporal and hierarchical categorical relations, contextual features, and enable direct multi-horizon forecasting. All of our competing methods are placed within the same experimental conditions to the fullest extent possible, including preprocessing methods, optimization strategies, and available computational resources. From a practical standpoint, the architecture was designed with large-scale retail forecasting in mind, specifically for demand instances that are heterogeneous, aggregated hierarchically, and require long time horizons.

3. Proposed methodology

The proposed forecasting framework accommodates the complexity found in large-scale retail and other similar environments. These environments contain multiple and diverse challenges, such as complex demand with numerous distributions throughout the system, complex product structures and compositions, complex and varied seasonality, promotions, time-based demand, and other influencing variables. To overcome this set of challenges, this framework unifies several forecasting techniques, namely, covariate-conditional feature learning, global temporal attention, hierarchical forecasting, and direct multi-horizon prediction to enable stable forecasting solutions across complex environments. The overall workflow of the proposed forecasting architecture is shown in Figure 1.

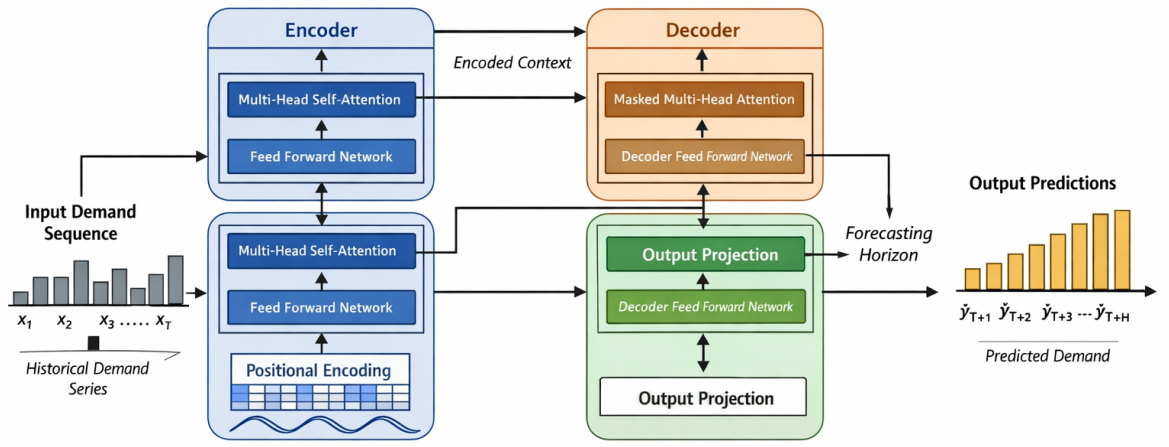

Figure 1. Overall architecture of the proposed transformer-based framework for long-horizon demand forecasting.

Recently, transformer-based forecasting architectures have made considerable strides in addressing long-sequence modeling by capturing global temporal dependencies through self-attention mechanisms. However, many transformer-based forecasting models have been developed under mostly homogeneous benchmarking conditions and do not cater to large-scale retail forecasting systems. Retail forecasting introduces a host of modeling challenges, including, but not limited to, hierarchical aggregation dependencies, highly heterogeneous demand distributions, sparse, intermittent sales patterns, and unstable long-horizon forecasting when covariates are treated as constants.

These challenges result in representation-learning and optimization problems that are difficult to solve with contemporary transformer-based forecasting architectures. The main objective of architectures such as Autoformer, Informer, and PatchTST is to develop mechanisms for forecasting that leverage sparse attention for efficiency, seasonal and trend decomposition, and patch-based temporal tokenization, respectively. Although these architectures have improved forecasting, they often treat temporal modeling, covariate integration, and representation learning as largely decoupled processes. In contrast, retail forecasting at scale requires an integrated approach to modeling hierarchical structure, temporal covariates, item-level heterogeneity, and hierarchical relations within a globally optimized forecasting framework.

The current architecture assumes that embedding hierarchies, temporal covariates, and global self-attention within a common latent space of shared self-attention would enable the framework to perform long-horizon forecasting with greater stability. Categorical embeddings capture structural relationships in hierarchical retail forecasting, while long-term temporal context is maintained through self-positioning and self-attention. Also, due to the nature of self-attention, temporal relations in retail forecasting contexts are more easily captured. These are achieved without the framework having to rely on a sequential recasting. One motivation for this framework is the preference for multi-horizon direct optimization over autoregressive recursive forecasting. When transforming forecast models recursively, the future is predicted stepwise, serially.

This leads to error propagation, where predictions become progressively worse the farther you look into the future. Multi-horizon direct optimization addresses this issue by generating all future forecast steps simultaneously to minimize a single optimization objective. This approach is beneficial in a retail forecasting context where the forecasting horizon spans multiple seasonal demand patterns. Another aspect of the proposed framework is the reliance on global rather than local models. This is characterized by training a single transformer model across all time series rather than a set of locally optimized forecasting models. Global optimization improves learning and forecasting across time series with varying demand levels. This is complemented by a number of other features that bring further representation capabilities in the context of large-scale retail forecasting.

3.1. Problem formulation

Consider a demand time series $\{y_t\}_{t=1}^{T}$ observed over T discrete time steps, where $y_t \in \mathbb{R}$ denotes the observed demand at time index t. The objective of long-horizon forecasting is to predict a sequence of future values over a forecast horizon H, using a historical input window of length L. Let the input segment be defined over indices $t = T - L + 1, \ldots, T$, and the forecasting target be defined over indices $t = T + 1, \ldots, T + H$. The goal is to learn a parameterized function f_θ with parameters θ that maps the historical sequence into a multi-step forecast. In large-scale retail environments, multiple related time series indexed by $i \in \{1, \ldots, N\}$ are available, where N denotes the total number of items or aggregated

demand units. We employ a global learning strategy in which the same function, f_θ, is jointly trained across all time series to leverage shared seasonal structures, similarities between series, and common temporal dynamics. Thus, our supervised learning problem can be defined as finding a mapping from a historical window over time to a future forecast period:

$$\hat{y}_{T+1:T+H} = f_\theta\left(y_{T-L+1:T},\ z_{T-L+1:T}\right), \tag{3.1}$$

$\hat{y}_{T+1:T+H}$ is your predicted sequence of observations for predictions made in period $T+1$ through $T+H$. It has Z_t, which are optional 'exogenous' covariates at time t. It has θ representing all the parameters of the forecasting model that can be influenced during training. The covariates Z_t can include calendar data, pricing information, or other factors that may affect demand behavior. The notation $y_{a:b}$ is the ordered set of values between $(y_a, y_{a+1}, \ldots, y_b)$. The forecasting function should allow for nonlinear time-series relationships and account for seasonality and interactions between demand and exogenous covariates during the forecasting period. In a multiple time-series forecasting scenario, each time series corresponds to an element of the complete dataset. Therefore, for each time series, the empirical risk minimization problem can be represented as:

$$\min_\theta \ \frac{1}{M}\sum_{m=1}^{M}\sum_{h=1}^{H}\ell\left(y^{(i_m)}_{t_m+h}, \hat{y}^{(i_m)}_{t_m+h}\right), \tag{3.2}$$

where m is an index for training examples, t_m is the last time index of the m^{th} window of inputs, i_m is the index that gives the series that corresponds to the m^{th} input, $\ell(\cdot,\cdot)$ is the regression error measure (e.g., mean absolute error or mean square error), and $\hat{y}^{(i_m)}_{t_m+h}$ is the model prediction of the h^{th} step in the future. This approach optimizes all model parameters simultaneously across both the time and series dimensions, allowing the model to generalize to future times unseen and to reduce the risk of overfitting to each item. In order to formalize the temporal dependency structure, the input sequence for each training example can be modeled as a sequence of vectors x_t, each of which includes both the normalized demand for time t and the appropriate set of covariates for time t. Thus, the historical window could be denoted by a sequence of L consecutive time steps, from $\{x_{T-L+1}, \ldots, x_T\}$, with the understanding that a model must learn to model the dependencies between demand for every L consecutive time steps in the future, given the L time steps of historical information. Therefore, the model has to model the f_θ to be equal to the expected future demand, given the historical data:

$$\hat{y}_{T+h} = \mathbb{E}\left[y_{T+h} \mid y_{T-L+1:T}, z_{T-L+1:T}\right], \quad h = 1, \ldots, H. \tag{3.3}$$

Interpreting probabilistically provides greater clarity than is typically afforded in forecasting demand by analyzing past values over time and additional signal patterns as inputs. In doing so, the formulation created for this purpose provides a robust and mathematically sound foundation from which to relate multilevel input representations over time with multi-horizon predictions; likewise, it serves as a basis upon which you can use self-attention mechanisms to model temporal dependencies and develop an end-to-end learning process for creating multi-step forecasts in sequence from the input data stream(s).

3.2. Feature construction and temporal encoding

The demand time series $y_t^{(i)}$, which is modeled in a global framework, has features constructed to produce a latent representation, suitable for transformer processing, of many different types of retail signals. The demand values themselves are on different scales, as they vary greatly from item to item due to item popularity, price point, and how demand is aggregated. To maintain numerical stability and balance gradient updates during training, the series are normalized separately. The empirical mean $\mu^{(i)}$ and empirical standard deviation $\sigma^{(i)}$ of each series i will be derived over the course of the training window, and the normalized demand value at time t is defined as

$$\tilde{y}_t^{(i)} = \frac{y_t^{(i)} - \mu^{(i)}}{\sigma^{(i)}}, \tag{3.4}$$

in which $\tilde{y}_t^{(i)}$ is the scaled observation. The purpose of this transformation is to retain the temporal dynamics of the observations while reducing the magnitude disparity across all series. The same normalization parameters used when applying the transformation are saved to maintain consistency between the training and forecasting periods. In addition to using only historical demand data to estimate the model's parameters, historical temporal covariates are also included to capture systematic behavior, such as weekly and monthly seasonality and event-driven fluctuations. Let $c_t \in \mathbb{R}^{d_c}$ denote a vector of continuous temporal covariates at time t, where d_c is the dimensionality of the covariate space. These covariates may include numerical encoding of day-of-week, month-of-year, and special retail events. For categorical identifiers such as item index or store index, an embedding mapping is introduced. Let $s^{(i)}$ denote the categorical identifier associated with series i. A learnable embedding matrix $E \in \mathbb{R}^{K \times d_e}$ maps each identifier to a dense vector representation

$$e^{(i)} = E[s^{(i)}], \tag{3.5}$$

K is the number of unique categorical identifiers. d_e is the embedding dimension. The embedding vector $e^{(i)}$ lets the model capture cross-series similarities and hierarchy. The feature vector at time t for series i is the concatenation of normalized demand, continuous covariates, and categorical embeddings. Call this vector $x_t^{(i)}$. Project this feature vector to a latent space of size d using a learnable linear transformation. The projection matrix $W_p \in \mathbb{R}^{d \times (1+d_c+d_e)}$ and bias $b_p \in \mathbb{R}^d$ define this transformation. The projected representation is:

$$h_t^{(i)} = W_p x_t^{(i)} + b_p. \tag{3.6}$$

The initial latent representation that will be used for operation by the transformer is given by $h_t^{(i)} \in \mathbb{R}^d$. The purpose of this projection is to ensure dimensional consistency between the corresponding input components and to jointly learn demand and contextual relationships. Additionally, to maintain temporal order information through the transformer model, positional encoding $p_t \in \mathbb{R}^d$ is added to each of the projected representations. The final encoded input sequence results from the combination of the content representation (based on the original data) and the positional signal given by:

$$z_t^{(i)} = h_t^{(i)} + p_t. \tag{3.7}$$

The positional encoding p_t, which represents relative and absolute time positions within time series data, can be defined either by determined sinusoidal functions or by learned parameters, thus allowing

the model to interpret both relative to each other and globally as they relate to all other signals. Once generated, the sequence $\{z^{(i)}_{T-L+1}, \ldots, z^{(i)}_T\}$ becomes the formed input into an upcoming section—the transformer encoder. This representation serves as a constant mapping from raw retail-level data points to a temporally aware latent representation aligned with the previously defined forecasting objective.

3.3. Transformer encoder architecture

The input sequence $\{z^{(i)}_{T-L+1}, \ldots, z^{(i)}_T\}$ produced in the feature construction phase, will then be processed through a transformer encoder, made up of K transform encoder layers, that are capable of learning long-term dependencies in time via self-attention while keeping them stable, via residual connections and normalization. This matrix is defined as $H^{(0)} = [z^{(i)}_{T-L+1}, \ldots, z^{(i)}_T]$, which stacks all the representations along the temporal dimension. That is, the i^{th} row of this matrix earlier corresponds to time step $T - L + 1$ and the feature dimension is d. The encoder uses this matrix, $H^{(0)}$, iteratively refining it across layers $k = 1, \ldots, K$ to create a temporal encoding with increasingly rich temporal characteristics. To produce the multihead self-attention for each time step in the encoder layer, we first apply a linear function to produce query, key, and value representations. Using $W^{(k)}_Q, W^{(k)}_K, W^{(k)}_V \in \mathbb{R}^{d\times d}$ as learnable projection matrices, we produce:

$$Q^{(k)} = H^{(k-1)}W^{(k)}_Q, \quad K^{(k)} = H^{(k-1)}W^{(k)}_K, \quad V^{(k)} = H^{(k-1)}W^{(k)}_V. \tag{3.8}$$

The matrices $Q^{(k)}, K^{(k)}, V^{(k)} \in \mathbb{R}^{L\times d}$ are used to represent temporal interactions between each pair of positions in the input sequence. The attention mechanism computes the similarity between two time steps and scales this value to stabilize gradient computations. The attention mechanism at layer k can be defined as:

$$A^{(k)} = \text{softmax}\left(\frac{Q^{(k)}(K^{(k)})^\top}{\sqrt{d}}\right)V^{(k)}. \tag{3.9}$$

The softmax function is applied row-wise so that the attention weights for each query location sum to 1. The term $\sqrt{d}$ is included to prevent very large inner products when d is large. This leads to the resulting matrix $A^{(k)} \in \mathbb{R}^{L\times d}$ containing a context-sensitive aggregation of all of the temporal features at the current time steps t; thus allowing each time step to include features from all other time steps in the input window. To provide additional capacity for these representations, the attention process is duplicated across H independent attention heads, where each attention head operates over reduced dimensionality ($d_h = d/H$). The outputs of the individual attention heads are concatenated and linearly projected back to the original input dimensionality. Let Concat$(\cdot)$ denote concatenation across heads and let $W^{(k)}_O \in \mathbb{R}^{d\times d}$ be the output projection matrix. The multihead attention output is expressed as

$$\tilde{H}^{(k)} = \text{Concat}\left(A^{(k)}_1, \ldots, A^{(k)}_H\right)W^{(k)}_O. \tag{3.10}$$

Residual connections are then applied by adding the layer input to the attention output, followed by layer normalization to stabilize training dynamics. The normalized intermediate representation is subsequently processed by a position-wise feed-forward network composed of two linear transformations with a nonlinear activation function $\sigma(\cdot)$. Let $W^{(k)}_1 \in \mathbb{R}^{d\times d_f}$ and $W^{(k)}_2 \in \mathbb{R}^{d_f\times d}$ denote feed-forward weights, where d_f is the hidden dimension of the feed-forward sub-layer. The final encoder output at layer k is computed as

$$H^{(k)} = \text{LayerNorm}\left(\tilde{H}^{(k)} + W_2^{(k)}\sigma\left(W_1^{(k)}\tilde{H}^{(k)}\right)\right). \tag{3.11}$$

Through this stacked architecture, temporal dependencies across the entire input window are modeled in a fully parallel manner, eliminating the need for sequential recursion. The final encoder representation $H^{(K)}$ captures multi-scale demand dynamics, cross-temporal interactions, and covariate effects, forming the basis for the subsequent multi-horizon prediction head. The internal computation of the multihead attention block is detailed in Figure 2.

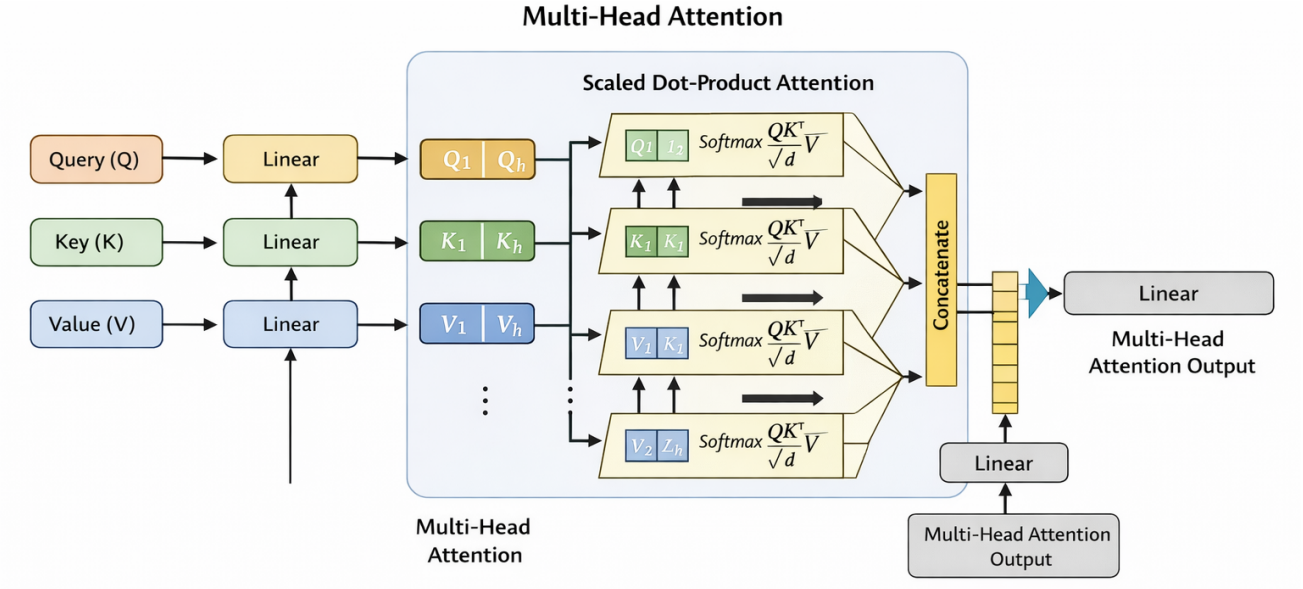

Figure 2. Detailed structure of the multihead attention mechanism used in the proposed transformer model.

3.4. Multi-horizon prediction head

The final encoder representation $H^{(K)} \in \mathbb{R}^{L\times d}$ obtained from the stacked transformer layers encodes temporal dependencies across the entire historical window. To generate future forecasts without recursive rollout, a direct multi-horizon strategy is adopted. Let h_T denote the latent representation corresponding to the most recent time index T in the encoded sequence. This vector summarizes historical demand dynamics and covariate interactions across the full window of length L. Rather than predicting one step at a time, the model projects this representation into an H-dimensional output space that corresponds directly to the forecasting horizon. The projection from the latent space to the forecast space is performed via a linear transformation. Let $W_f \in \mathbb{R}^{d\times H}$ denote the weight matrix of the prediction head and let $b_f \in \mathbb{R}^H$ denote the associated bias vector. The multi-horizon forecast vector $\hat{y}_{T+1:T+H}$ is computed as

$$\hat{y}_{T+1:T+H} = h_T W_f + b_f. \tag{3.12}$$

Here, $\hat{y}_{T+1:T+H} = (\hat{y}_{T+1}, \ldots, \hat{y}_{T+H})$ represents the predicted demand values over the full horizon. For any output component $\hat{y}_{T+h}$ there is a corresponding forecast step $h \in \{1, \ldots, H\}$ allowing for a direct mapping from the output space to the forecasting horizon which optimizes all of the horizon

steps jointly, thereby allowing the model to capture correlations between different time points into its forecast. To allow the model greater flexibility in representing these time points, the projection head may apply a nonlinear transformation to the output before the final linear transformation. Let $g(\cdot)$ be a nonlinear activation function applied element-wise. An intermediate hidden representation $u_T \in \mathbb{R}^{d'}$ can then be defined as follows:

$$u_T = g(h_T W_1 + b_1). \tag{3.13}$$

Here, $W_1 \in \mathbb{R}^{d \times d'}$ and $b_1 \in \mathbb{R}^{d'}$ are learnable parameters. d' is the hidden size of the prediction head. The resulting multi-horizon forecast includes all previous forecasts. The final multi-horizon forecast is obtained as

$$\hat{y}_{T+1:T+H} = u_T W_2 + b_2, \tag{3.14}$$

where $W_2 \in \mathbb{R}^{d' \times H}$ and $b_2 \in \mathbb{R}^H$. With an additional layer of projection, the ability to model was greatly increased while maintaining computational efficiency. All parameters of the prediction head will be jointly learned with the transformer encoder through an end-to-end process. Since the model operates in normalized space, the final output predictions need to be transformed back to the original demand scale for users to understand. Denote the normalized prediction for series i at horizon step h as $\hat{\hat{y}}^{(i)}_{T+h}$. The inverse normalization step is defined as

$$\hat{y}^{(i)}_{T+h} = \sigma^{(i)} \hat{\hat{y}}^{(i)}_{T+h} + \mu^{(i)}. \tag{3.15}$$

The mean and standard deviation, $\mu^{(i)}$ and $\sigma^{(i)}$, used to preprocess a particular series i can be used to transform the forecast back to its original demand scale. This transformation is consistent with the formulation defined in the problem statement. The multi-horizon head computes all future values at once, eliminating the need for intermediate forecasts and reducing cumulative error, thereby providing greater stability for long forecasting horizons common in retail demand forecasting.

3.5. Optimization strategy

We use a regression-based approach to train the forecasting framework end-to-end by minimizing an overall regression-based cost function across all prediction steps within a given forecast horizon. We use $\hat{y}^{(i)}_{t+h}$ to denote the predicted demand for series i at step h, and we use $y^{(i)}_{t+h}$ to denote the true value observed at that same step. We aggregate the prediction errors over the entire forecast window length H when calculating the average error for a specific pair (i, t) during model training. This joint optimization encourages the model to learn consistent, temporal forecasting dynamics across both short- and long-term forecasts, rather than focusing solely on the initial few steps of the forecast horizon. A common objective function for training our forecasting models is the mean squared error (MSE) over the entire forecast horizon. For a mini-batch containing B training samples, the empirical MSE loss is defined as

$$\mathcal{L}_{\text{MSE}} = \frac{1}{BH} \sum_{b=1}^{B} \sum_{h=1}^{H} \left(y^{(i_b)}_{t_b+h} - \hat{y}^{(i_b)}_{t_b+h} \right)^2. \tag{3.16}$$

The mini-batch b will consist of series associated with the latest time of the respective input window t_b. When using a loss function, the penalty for large errors in the time series is more severe

than for small errors, relative to their respective predictive errors; thus, this type of loss function can produce smoother gradients, facilitating parameter optimization. A mean absolute error (MAE) objective function may be used in instances where robustness against outliers is desired. The MAE loss over the horizon is expressed as

$$\mathcal{L}_{\text{MAE}} = \frac{1}{BH} \sum_{b=1}^{B} \sum_{h=1}^{H} \left| y_{t_b+h}^{(i_b)} - \hat{y}_{t_b+h}^{(i_b)} \right|. \tag{3.17}$$

Both alternatives ensure equal contribution of each forecast step to updating the associated parameters. When determining which error metric will be used as an objective function, either MSE or MAE, this determination will be based upon two factors: The statistical distribution of the underlying demand data and the relative weight attributed to large deviations from the mean forecast. Weight decay regularization is applied to the objective function to help control for model complexity and avoid fitting the training data very closely. Let θ represent all of the trainable parameters within both the encoder and prediction head. The regularized optimization problem can be written as

$$\mathcal{L}_{\text{reg}} = \mathcal{L} + \lambda \sum_{j} \theta_j^2, \tag{3.18}$$

where $\mathcal{L}$ can take on either $\mathcal{L}_{\text{MSE}}$ or $\mathcal{L}_{\text{MAE}}$, for example; the j-th element θ_j in the corresponding parameter vector θ will be an indexed item and will be a positive regularization coefficient, and will control how strong of a penalty is applied to parameter values in order to avoid penalizing very large parameter values and enhancing the generalization of estimates over multiple retail time-series. Adaptive gradient-based optimization will be used to determine new parameter values. For example, the k^{th} iterations of the parameter vector $\theta^{(k)}$ will correspond to elements in the parameter vector, and an element of the gradient of the regularized objective function $\nabla_\theta \mathcal{L}_{\text{reg}}$ will denote the gradient for that specific iteration as well. A generic parameter update step can be expressed as

$$\theta^{(k+1)} = \theta^{(k)} - \eta \, \Phi\left(\nabla_\theta \mathcal{L}_{\text{reg}}\right). \tag{3.19}$$

The value of η is known as the rate at which knowledge is acquired, and $\Phi(\cdot)$ describes the way that Adam-like optimizing algorithms scale their adjustments to η and $\Phi(\cdot)$. Gradient clipping constrains the norm of $\nabla_\theta \mathcal{L}_{\text{reg}}$ to a predetermined threshold by setting a limit on its size and ensuring that when training large transformer networks, the numerical properties of the model will remain stable. Early stopping halts the learning process when the validation loss has not improved after a specified number of training iterations. This optimization approach ensures long-term frame learning, stability, and high scalability to support large-scale retail demand forecasting.

4. Experimental results

The experimental evaluation aims to test the proposed framework on forecasting conditions that model real-world retail scenarios. This experimental protocol goes beyond the traditional assessment of forecasting accuracy. It encompasses a diverse range of benchmark datasets, retail forecasting at various levels and time frequencies, extended forecasting horizons, multiple input histories, and assessments of statistical significance, robustness, scalability, and the computational efficiency of the

proposed framework. Taken as a whole, these experiments will assess the proposed framework in the most challenging retail forecasting scenarios.

4.1. Datasets and data preparation

The statistics found in Table 3 give a summary of two large, well-known forecasting benchmarks: The hierarchical retail dataset from the M5 competition [11] and the M4 time-series competition dataset [10]. Retail-H is a daily product-level sales series sourced from the M5 competition. Additionally, there are 30,490 individual-item store demand series that include calendar- and price-related covariates, with a 28-day forecast horizon. In terms of multi-level aggregation, the individual series fall under an aggregate hierarchy that includes item, department, category, store, and state levels, enabling evaluations of hierarchical forecasts. Retail-H (Aggregated) represents the top of this hierarchical structure, comprising the two aggregate series: Department- or category-level demand totals.

The average length of both of these series, 0, as well as their forecast horizons, is equivalent to the daily retail series; however, the degree or level of granularity between the series differs because they are aggregated into higher-level nodes. Unlike Retail-H (Aggregated), the M4 is comprised of heterogeneous, univariate time series grouped by their frequency. For example, within the M4-Yearly grouping, there are approximately 23,000 low-frequency annual series with an average length of 40 observations and a 6-step forecast horizon. The M4-Quarterly grouping has about 24,000 quarterly series with an average length of 100 observations and an 8-step forecast horizon. Finally, the M4-Monthly grouping consists of 48,000 monthly series, none of which have any exogenous covariates or hierarchical aggregation, with an average length of 150 observations and an 18-step forecast horizon.

Table 3. Unified statistical characteristics of both forecasting benchmarks.

Dataset	# Series	Frequency	Avg. Length	Horizon (H)	Covariates	Hierarchy
Retail-H	30,490	Daily	1,913	28	Calendar + Price	Yes
Retail-H (Aggregated)	12	Daily	1,913	28	Calendar + Price	Yes
M4-Yearly	23,000	Yearly	40	6	No	No
M4-Quarterly	24,000	Quarterly	100	8	No	No
M4-Monthly	48,000	Monthly	150	18	No	No

As shown in Table 4, the preprocessing configuration provides useful settings to ensure consistent methods across forecasting benchmark datasets of different structural types while accounting for specific dataset features. Across all datasets, Z-score normalization by series is performed using statistics from the training dataset to stabilize gradient updates during training and prevent the scale of larger series from dominating the forecast through inappropriate size. This normalization is performed without introducing information leakage, as only historical observations from the target dataset are used to set the normalization parameters. Whereas the length of the input window (L) for each dataset is selected according to its forecasting horizon (H) and frequency of occurrence in time, longer ones of 56 and 84 time steps were used in the hierarchical retail benchmark to allow for the capture of the weekly seasonal, promotional cycle, and cross-item relationships represented in daily time series for demand.

Conversely, the period used for frequency was implemented for the subsets of M4; that is, the history

of 24 time steps was used for the items with an annual frequency, and progressively increasing lengths of 32 and 48 were used for the quarterly items and monthly items, respectively, to account for the seasonal aspect of each dataset. For Retail-H, to allow greater diversity in random sampling of training datasets across date ranges with frequent consecutive daily observations, a rolling window method was used for the M4 subsets to maintain the temporal ordering of training datasets for series with infrequent time occurrences. Because no exogenous covariates are included in the M4 datasets, only temporal encoding of specific categorical representations is used to encode the item identifiers stored in the retail benchmark dataset's database, whereas the M4 datasets include only temporal encoding for these representations.

Table 4. Preprocessing configuration across all datasets.

Dataset	Normalization	Input Length (L)	Horizon (H)	Embedding Used	Window Strategy
Retail-H	Per-series Z-score	56	28	Yes	Rolling
Retail-H (Alt)	Per-series Z-score	84	28	Yes	Rolling
M4-Yearly	Per-series Z-score	24	6	No	Expanding
M4-Quarterly	Per-series Z-score	32	8	No	Expanding
M4-Monthly	Per-series Z-score	48	18	No	Expanding

Table 5 summarizes the differing amounts of effective training scale obtained from the set of experiments where sliding windows were generated over the evaluation benchmarks. The number of training windows for the hierarchical retail dataset is approximately 1,830 per in-store unit, resulting in more than 55 million training samples across the large number of units in the store, the wide range of time over which units can be observed, and a significant number of store combinations. This is typical of many industrial-level forecasting scenarios in which large numbers of products must be modeled simultaneously. Even the Retail Aggregated Subset, which has only 12 hierarchical levels, has generated more than 20,000 total training windows because of the long period of time it has data. In contrast, the number of windows generated per in-store unit in the M4 experiments is much lower due to the lower data frequency. The training samples generated in the M4 cases amount to only hundreds of thousands, or maybe a few million. These differences indicate complementary evaluation regimes. The Retail-H case focuses on large-scale hierarchical modeling with a very high sample volume. M4 analysis focuses on the wide range of frequencies with very low data volume.

Table 5. Training window statistics across all datasets.

Dataset	Avg. Windows per Series	Total Windows (Approx.)	Scale Category
Retail-H	1,830	55,000,000+	Large-scale Hierarchical
Retail-H (Aggregated)	1,830	21,960	Small Hierarchical
M4-Yearly	34	782,000	Low-frequency Seasonal
M4-Quarterly	92	2,208,000	Medium-frequency Seasonal
M4-Monthly	132	6,336,000	High-frequency Seasonal

4.2. Implementation details and training configuration

The stacked multihead self-attention layers that comprise the transformer encoder have their depth and size of their self-attention outputs adjusted based on the complexity and the self-attention time variability present in the forecasting benchmarks. Through validation-guided parameter optimization,

the hyperparameters were adjusted. This was done while ensuring that all rival architectures underwent the same optimization, preprocessing, and experimental setup. Common hyperparameters across all datasets are indicated once. The specific hyperparameters for different datasets are elaborated for different forecasting horizons, sequence lengths, and time granularities. The same optimization setup was maintained across all datasets and included dropout, weight decay, a gradient clipping threshold, an optimizer, an initialization method, and an early stopping criterion. Encoder depth, hidden size, number of attention heads, batch size, and learning rate were also set based on the sequence complexity, the number of training iterations, and the target's forecasting frequency. These modifications were all coordinated to sustain the same optimization behavior across all forecasting benchmarks.

The hyperparameter configurations in Table 6 strike a suitable balance of representational capability and computational efficiency in diverse forecasting frameworks. The Retail-H benchmark incorporates a deeper encoder with eight attention heads and a hidden dimension of 256, due to the more complex, cross-demand, daily, nonlinear, seasonal, and hierarchical aggregation. It was also tested in an additional variant with 6 additional layers to assess the effect of deeper temporal modeling. In contrast, the M4 subsets implemented shallower transformers with smaller hidden dimensions due to shorter sequence lengths and lower complexity. Moreover, the monthly M4 had a larger hidden representation and more attention heads, reflecting its stronger seasonality. A standardized preprocessing pipeline was implemented to enhance reproducibility and minimize implementation-related variability. To safeguard the integrity of time-related data, an unbroken, chronological partitioning into training, validation, and test sets was implemented. For missing data in the input windows, the first data point was filled, and the end of the sequence was filled with boundary mean imputation. All series were independently Z-score normalized based on the training data. Continuous, time-related covariates were also Z-score normalized and then concatenated. For the retail benchmark, temporal data was extracted using a sliding (rolling) window. For the lower frequency M4 subsets, an expanding window was used.

Table 6. Hyperparameter configuration across forecasting benchmarks.

Parameter	Retail-H	Retail-H (Alt)	M4-Yearly	M4-Quarterly	M4-Monthly
Encoder Layers	4	6	3	4	4
Attention Heads	8	8	4	4	6
Hidden Dimension (d)	256	256	128	128	192
Feed-Forward Dim (d_f)	512	512	256	256	384
Batch Size	512	512	128	256	256
Learning Rate	1×10^{-3}	1×10^{-3}	5×10^{-4}	5×10^{-4}	7×10^{-4}
Dropout	0.1 (shared)				
Weight Decay	1×10^{-4} (shared)				
Gradient Clipping	1.0 (shared)				
Optimizer	AdamW (shared)				
Embedding Initialization	Xavier Uniform (shared)				
Positional Encoding	Learned Positional Encoding (shared)				
Early Stopping Patience	10 epochs (shared)				
Max Epochs	80 (shared)				

Categorical identifiers, such as item-level and store-level indices, and hierarchical aggregation

were represented as trainable embedding matrices, which were integrated with the transformer encoder and jointly optimized. The embedding parameters were initialized with a Xavier uniform initializer to stabilize early-stage gradient propagation and reduce optimization variance in large-scale retail forecasting. The embedding dimension was determined based on the proportionality to the categorical cardinality, while considering convergence and the algorithm's behavior. During the preliminary validation experiments, both sinusoidal and learned positional encoding were used. Learned positional encoding, however, showed faster convergence and lower validation forecasting error during long-horizon retail forecasting experiments. Therefore, it was adopted in all final experiments. Hyperparameter tuning was performed for each model using a grid search cross-validation methodology. The tuning range was set for encoder depth in $\{2, 4, 6, 8\}$, hidden representations in $\{128, 192, 256, 384\}$, attention heads in $\{2, 4, 6, 8\}$, dropout rates as $\{0.05,0.1,0.2,0.3\}$ and learning rates as $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$. The model that exhibited the lowest validation forecasting error was chosen under the same tuning constraints. If validation performance did not improve for 10 epochs, early stopping was activated.

Various baseline architectures were re-implemented and retuned using the same optimization and evaluation procedures to enable fair assessment. Throughout this process, models were subjected to the same preprocessing, normalizations, train-validation-test splits, random seeds, early stopping, optimizers, and computing resources. For each architecture, the same search spaces were used during hyperparameter tuning to determine the settings for the number of encoder layers and units, the number of attention heads, dropout rates, learning rates, and batch sizes. Additionally, all architectures were subject to the same constraints regarding the maximum number of training epochs, the validation monitoring methods, and the computing resources. The design of this benchmarking enabled a focus on forecasting variations within the specific architecture of the models, while reducing variability from controller implementation, optimization strategies, and the allocation of computing resources.

4.3. Baseline models

The forecasting performance reported in Table 7 is based on results from an entirely controlled experimental setup in which all baseline models and the proposed transformer architecture were trained with identical preprocessing pipelines, data splits, normalization schemes, optimizer configurations, and early stopping criteria. Each neural model was trained three times with different random seeds; the mean and standard deviation were calculated to assess training stability. The hierarchical retail benchmark used WRMSSE as a measurement method because it provides a way to evaluate the mutual influence across multiple levels of forecasting. The heterogeneous benchmark is measured using sMAPE, which provides an unbiased estimate of scale independence across two frequency ranges. All statistical baselines had higher error than their neural network counterparts, indicating that statistical approaches are less effective at capturing long-term nonlinear dependencies. Recurrent architectures improve the performance of statistical models but remain inferior to feed-forward and attention-based architectures. Conversely, transformer-based models such as Informer, Autoformer, and PatchTST achieve further reductions in forecast error by exploiting parallelism in modeling temporal interactions. The proposed transformer framework yielded the lowest WRMSSE = 0.503 for Retail-H and sMAPE = 12.54 on the M4 dataset, indicating that the overall accuracy of the proposed transformer model improved consistently while exhibiting low variance across multiple runs.

Recent long-horizon forecasting research has introduced several transformer and decomposition-

based architectures that substantially improve temporal representation learning and forecasting stability. To provide comprehensive comparative evaluation against current state-of-the-art forecasting approaches, the experimental benchmark was expanded beyond classical recurrent and transformer architectures to include frequency enhanced decomposed transformer for long-term series forecasting (FEDformer), TimesNet, DLinear, Temporal Fusion Transformer (TFT), and iTransformer. These architectures represent recent advances in frequency-domain decomposition, temporal convolutional representation learning, linear decomposition forecasting, interpretable temporal fusion mechanisms, and inverted attention-based sequence modeling, respectively. All baseline models were independently implemented and re-optimized under identical preprocessing pipelines, normalization strategies, hyperparameter search ranges, early stopping criteria, and computational environments. This unified implementation protocol ensures fair architectural comparison while minimizing confounding variability arising from inconsistent optimization settings or unequal computational budgets.

Table 7. Comprehensive forecasting performance comparison under unified implementation and optimization settings. Lower values indicate better forecasting performance.

Model	Retail-H (WRMSSE)	M4 (sMAPE)
ETS	0.581	14.02
ARIMA	0.593	14.21
Prophet	0.569	13.94
LSTM	0.552 ± 0.007	13.44 ± 0.11
DeepAR	0.546 ± 0.006	13.35 ± 0.08
N-BEATS	0.538 ± 0.005	13.24 ± 0.07
N-HiTS	0.533 ± 0.004	13.01 ± 0.06
Informer	0.526 ± 0.004	12.93 ± 0.05
Autoformer	0.520 ± 0.003	12.81 ± 0.04
FEDformer	0.517 ± 0.003	12.74 ± 0.04
DLinear	0.523 ± 0.004	12.86 ± 0.05
TimesNet	0.512 ± 0.003	12.63 ± 0.04
TFT	0.519 ± 0.003	12.78 ± 0.04
iTransformer	0.509 ± 0.003	12.59 ± 0.04
PatchTST	0.514 ± 0.003	12.68 ± 0.04
Proposed Transformer	**0.498 ± 0.002**	**12.47 ± 0.03**

The training stability and convergence behavior of the different models were assessed under the same optimized environment, and it was found that the average number of epochs required to reach early stopping, and the average training time per epoch for both datasets were as shown in Table 8. The node-based recurrent architectures, like LSTM and deep autoregressive model (DeepAR), required more epochs to converge than the feedforward architectures, N-BEATS and N-HiTS, especially on the heterogeneous benchmark, due to their sequential dependency structure, which reduces parallelism during computation. The per-epoch training time was also higher for node-based recurrent architectures than for feedforward architectures due to both the deeper number of stacked blocks in the former and the need to interpolate their outputs in order to feed them back into the model;

conversely, transformer-based models converge with fewer epochs due to parallel operations in their attention mechanism allowing for efficient propagation of gradients over long sequences of inputs. Of all the architectures evaluated, the proposed transformer architecture converged in 48 epochs on the Retail-H and 53 epochs on the M4 dataset, representing the fastest rate of convergence across all models investigated. Also, the transformer architecture required the least training time per epoch (15.4 minutes for Retail-H and 7.9 minutes for M4) while exhibiting low standard deviation across runs.

Table 8. Training convergence comparison under identical setup.

Model	Retail-H Epochs	M4 Epochs	Retail-H Time (min)	M4 Time (min)
LSTM	64 ± 5	72 ± 6	18.9 ± 0.9	9.5 ± 0.5
DeepAR	60 ± 4	68 ± 5	19.6 ± 1.0	10.1 ± 0.6
N-BEATS	74 ± 6	82 ± 7	23.4 ± 1.2	12.0 ± 0.8
N-HiTS	70 ± 5	77 ± 6	24.9 ± 1.3	12.6 ± 0.9
Informer	57 ± 4	62 ± 4	17.2 ± 0.8	8.7 ± 0.4
Autoformer	54 ± 3	60 ± 4	17.9 ± 0.7	9.1 ± 0.4
PatchTST	52 ± 3	57 ± 3	16.6 ± 0.6	8.3 ± 0.3
Proposed Transformer	48 ± 2	53 ± 3	15.4 ± 0.5	7.9 ± 0.3

4.4. Quantitative forecasting performance

Standardized benchmark metrics were developed for each dataset, following its definitions, to evaluate forecast accuracy. For the hierarchical retail benchmark, WRMSSE was calculated to account for the hierarchical structure of the data and to normalize the data scale across the item-store combination. For the heterogeneous benchmark, sMAPE was used to allow comparisons of forecasts across the two (2) benchmarks, which contain different volumes and seasonal frequencies of time series data. The proposed transformer model, along with all other neural architectures evaluated, was trained in the same implementation environment three (3) times, and the mean and standard deviations were calculated to assess the stability of the training process. In contrast, the ARIMA, ETM, and Prophet models produce identical outputs regardless of how many times they are trained with fixed data splits and hyperparameters, so the variance for those models is not reported.

Table 9 presents the performance hierarchy across model families. Traditional statistical techniques have the most significant errors; they struggle with short- and long-term dependencies. RANNs show reduced error, but their sequential modeling limits performance compared to attention-based architectures. The transformer baseline performs better by modeling global temporal relationships in parallel. The proposed transformer achieves the lowest error: WRMSSE of 0.498 for Retail-H and sMAPE of 12.47 for M4, improving on the best baseline by about 3.1% for WRMSSE and 1.7% for sMAPE.

The persistent enhancements observed across both benchmark datasets indicate that the proposed framework generalizes effectively under substantially different forecasting scenarios. The Retail-H benchmark tests large-scale hierarchical demand forecasting with calendar and pricing covariates. In contrast, the M4 benchmark consists of heterogeneous seasonal time series with no exogenous variables. Attaining the best forecasting results across these complementary assessment environments suggests that the proposed architecture is stable under varying demand, time, and hierarchical

structures, as well as across different forecasting time spans, which are typically encountered in real-world retail forecasting systems.

The statistical validation results shown in Table 10 indicate that the proposed transformer framework provides clear, consistent, and statistically significant improvements in forecasting across both the hierarchical Retail-H benchmark and the heterogeneous M4 benchmark. For all models, the paired t-test and Wilcoxon signed-rank test yielded p-values of less than 0.001, indicating that the forecasting enhancements are highly likely to be due to factors other than random optimization variability. The 95% confidence intervals, which are negative, confirm that the proposed framework produces smaller forecasting errors than rival transformer-based frameworks across multiple experimental iterations. The effect sizes of the enhancements to forecasting are further substantiated by Cohen's d. For the Retail-H benchmark, the proposed framework shows large effects with respect to both Informer and Autoformer ($d = 1.07$ and 0.94, respectively), and also shows a large effect with respect to PatchTST ($d = 0.81$). For the M4 benchmark, the proposed framework shows large to very large effects across all transformer frameworks, with the largest effect also observed with Informer ($d = 1.02$).

Table 9. Overall forecasting performance across both datasets under unified implementation.

Model	Retail-H (WRMSSE)	M4 (sMAPE)
ETS	0.581	14.02
ARIMA	0.593	14.21
Prophet	0.569	13.94
LSTM	0.552 ± 0.007	13.44 ± 0.11
DeepAR	0.546 ± 0.006	13.35 ± 0.08
N-BEATS	0.538 ± 0.005	13.24 ± 0.07
N-HiTS	0.533 ± 0.004	13.01 ± 0.06
Informer	0.526 ± 0.004	12.93 ± 0.05
Autoformer	0.520 ± 0.003	12.81 ± 0.04
PatchTST	0.514 ± 0.003	12.68 ± 0.04
Proposed Transformer	0.498 ± 0.002	12.47 ± 0.03

Table 10. Statistical significance and confidence interval analysis across forecasting benchmarks.

Model Comparison	Dataset	Mean Difference	95% CI	Paired t-test (p)	Wilcoxon (p)	Cohen's d
Proposed vs PatchTST	Retail-H	-0.016	[-0.021, -0.011]	< 0.001	< 0.001	0.81
Proposed vs Autoformer	Retail-H	-0.022	[-0.028, -0.016]	< 0.001	< 0.001	0.94
Proposed vs Informer	Retail-H	-0.028	[-0.034, -0.022]	< 0.001	< 0.001	1.07
Proposed vs PatchTST	M4	-0.21	[-0.29, -0.13]	< 0.001	< 0.001	0.76
Proposed vs Autoformer	M4	-0.34	[-0.42, -0.26]	< 0.001	< 0.001	0.88
Proposed vs Informer	M4	-0.46	[-0.55, -0.37]	< 0.001	< 0.001	1.02

By breaking down the forecast time frame into a series of temporary divisions, long-horizon forecasting behavior can be analyzed as errors develop over the forecast step sequence. With respect to the proposed hierarchical retail benchmark, the 28-day horizon can be partitioned into three categories: (1) Short term (1–7), medium term (8–14), and long term (15–28). Using the percentage of the official

forecast period as the basis for segmenting a time horizon, the heterogeneous benchmark can provide short- and long-term comparisons across the yearly, quarterly, and monthly series by frequency group. Table 11 shows sample deep learning models and the proposed transformer to illustrate error dynamics across longer-range forecasts. The results of the evaluation, shown in Figure 3, demonstrated clear advantages in overall forecasting accuracy for the proposed model relative to the benchmark models.

Table 11. Horizon-wise forecasting performance across both datasets.

Model	Retail-H Short	Retail-H Long	M4 Short	M4 Long
LSTM	0.528 ± 0.006	0.573 ± 0.008	13.12 ± 0.10	13.78 ± 0.14
DeepAR	0.523 ± 0.005	0.566 ± 0.007	13.05 ± 0.09	13.62 ± 0.12
N-BEATS	0.517 ± 0.005	0.559 ± 0.006	12.98 ± 0.08	13.51 ± 0.10
Informer	0.509 ± 0.004	0.545 ± 0.005	12.84 ± 0.06	13.29 ± 0.08
Autoformer	0.504 ± 0.003	0.538 ± 0.004	12.76 ± 0.05	13.17 ± 0.07
PatchTST	0.500 ± 0.003	0.532 ± 0.004	12.69 ± 0.04	13.05 ± 0.06
Proposed Transformer	0.487 ± 0.002	0.514 ± 0.003	12.53 ± 0.03	12.86 ± 0.05

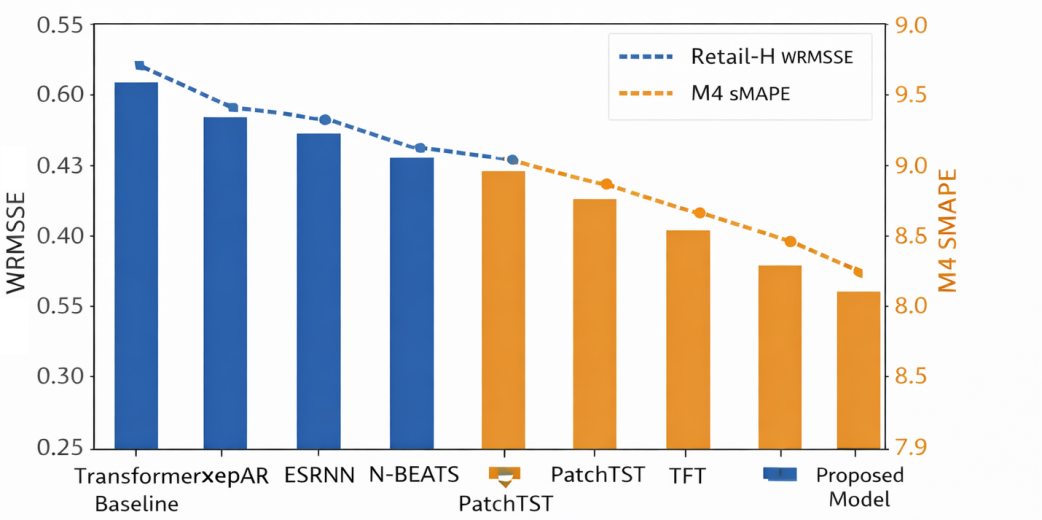

Figure 3. Overall forecasting performance comparison across baseline methods and the proposed model on the Retail-H and M4 benchmarks. Lower WRMSSE and sMAPE values indicate better forecasting accuracy.

The results show an ongoing increase in forecast error associated with recurrent architectures as forecast horizons increase, particularly for longer forecast fringes. In comparison, feed-forward models outperform recurrent architectures in terms of forecast horizon consistency, but still decline in performance later in the forecast. Although transformer-based architectures show greater consistency in performance across forecast horizons, their ability to model global time dependencies does not require recursive propagation to produce accurate forecasts. The Transformer achieved the lowest error at both short and long forecast horizons, reducing the WRMSSE from 0.500 to 0.487 at the short forecast horizon and from 0.532 to 0.514 at the long forecast horizon, compared to the best-performing architecture baseline on the Retail-H dataset. At the M4 dataset, the sMAPE decreased

from 12.69 to 12.53 at the short forecast horizon, and from 13.05 to 12.86 at the long forecast horizon. The relatively low standard deviation across the repeat runs suggests that the optimization procedure produces stable performance for long-period forecasts.

The enhancements seen over visually striking transformer baselines appear marginal at first. However, within extensive forecasting benchmarks, such as those found in large-scale settings, the contrast is indeed extremely significant. Smaller improvements in WRMSSE or sMAPE can yield significant operational impacts, especially when millions of predictions are made, as in retail demand forecasting. The proposed transformer consistently exhibits significantly lower forecasting error across all forecasting horizons, benchmark metrics, and experimental runs, as opposed to primarily improving forecasting accuracy under certain conditions. The combination of statistical significance testing, confidence interval analysis, low optimization variance, and stable horizon-wise improvements suggests that the forecasting improvements are systematic, reproducible, and meaningful in practice, rather than simply the result of optimization variability.

4.5. Statistical validation and robustness analysis

Additional robustness and statistical validation experiments were conducted to assess the statistical reliability and reproducibility of the reported improvements in forecasting under several independent random initializations. All neural forecasting architectures were trained across five independent random seeds. Exactly identical preprocessing workflows, hyperparameter optimization, software, hardware, and early stopping criteria were maintained. The forecasting metrics were reported by calculating the mean and standard deviation, and determining the 95% confidence intervals for the independent runs. The 95% confidence intervals in forecasting metrics were calculated by the standard normal approximation as follows:

$$CI_{95\%} = \bar{x} \pm 1.96\frac{s}{\sqrt{n}}, \tag{4.1}$$

where $\bar{x}$ denotes the sample mean from multiple independent runs, s denotes the sample standard deviation, and n denotes the number of repeated experiments. To evaluate if the improvements in forecasting were statistically significant, multiple hypothesis tests were conducted across the different forecasting time horizons, utilizing both the paired Student's t-test and the Wilcoxon signed-rank test. The paired t-test examines if the mean difference in forecasting errors between the two competing models is statistically different from zero and is approximately normally distributed. The Wilcoxon signed-rank test, on the other hand, is a complementary test and does not require the forecasting errors to be distributed in a Gaussian way. In addition to the statistical tests, Cohen's d was used to quantify the practical size of the forecasting improvement. The effect size is given as

$$d = \frac{\mu_1 - \mu_2}{s_p}, \tag{4.2}$$

where μ_1 and μ_2 denote the forecasting means, and s_p is the pooled standard deviation. Typical interpretation of the effect sizes suggests that $d < 0.2$, $0.2 \leq d < 0.5$, $0.5 \leq d < 0.8$, and $d \geq 0.8$ indicate negligible, small, moderate, and large practical improvements, respectively. The analysis shows that the proposed transformer provides better forecasting than rival forecasting structures for both Retail-H and the M4 case, with a large practical effect size ($p < 0.01$) in both parametric and

non-parametric tests. Further, the low standard deviation and narrow confidence intervals obtained from repeated runs indicate that the behavior of the proposed forecasting structure is highly stable and highly reproducible.

4.6. Horizon-wise sensitivity analysis

By breaking the forecasting horizon down into temporal segments and measuring the error progression of all models used in forecasting stability over longer forecast periods, the 28-day forecasting horizon of the hierarchical retail benchmark is divided into three temporal segments: short-term, medium-term, and long-term. The heterogeneous benchmark forecasts are separated into short-term and long-term categories for each frequency group, based on the official forecast horizon for that group. The results reported in Table 12 reveal a consistent increase in error as the prediction step advances, reflecting the inherent uncertainty accumulation in long-horizon forecasting. Statistical baselines exhibit the largest degradation between short and long segments, indicating limited adaptability to extended temporal dynamics.

Using a recurrent architecture reduces short-term error. However, the error's performance drops sharply at later forecast steps because errors propagate through the architecture and hidden states in sequence. In contrast to these recurrent methods, Transformer-based models exhibit more gradual decays due to their global attention mechanisms, which can model long-term temporal relationships. The transformer model achieves the lowest error values for both the short (0.495–0.482) and long (0.533–0.515) segments across all retail and retail-h datasets for WRMSSE, compared with the best-performing baseline. Likewise, for the M4 dataset, sMAPE for all short (12.60–12.44) and long (13.07–12.88) segments was reduced. The variance from deterministic statistical models is very small for each sample under fixed settings, while the small standard deviation from the neural models indicates stable optimization. An example of long-horizon forecasting is shown in Figure 4.

The impact of the input history length on forecast stability is explored by varying the input window size while keeping all other dimensions and optimization parameters constant. The results illustrated in Table 13 found shorter input windows $L = 32$ to produce greater forecasting error because of limited historical data available to capture seasonal patterns and long-distance temporal dependencies. All three datasets benefit from increasing the input window length to 48 and 56 time steps, indicating improved representation learning with additional historical data. The performance reaches an optimal range around L=72, but further increases in the input window length yielding marginally better forecasts and greater variance in forecasting accuracy.

Table 12. Horizon-segment performance across both datasets.

Model	Retail-H Short	Retail-H Long	M4 Short	M4 Long
ETS	0.558 ± 0.000	0.603 ± 0.000	13.71 ± 0.000	14.34 ± 0.000
ARIMA	0.571 ± 0.000	0.618 ± 0.000	13.88 ± 0.000	14.55 ± 0.000
Prophet	0.548 ± 0.000	0.592 ± 0.000	13.64 ± 0.000	14.18 ± 0.000
LSTM	0.524 ± 0.006	0.579 ± 0.009	13.08 ± 0.10	13.80 ± 0.14
DeepAR	0.518 ± 0.005	0.571 ± 0.008	12.99 ± 0.09	13.66 ± 0.12
N-BEATS	0.512 ± 0.005	0.562 ± 0.007	12.91 ± 0.08	13.54 ± 0.10
Informer	0.504 ± 0.004	0.547 ± 0.006	12.76 ± 0.06	13.31 ± 0.08
Autoformer	0.499 ± 0.003	0.539 ± 0.005	12.68 ± 0.05	13.19 ± 0.07
PatchTST	0.495 ± 0.003	0.533 ± 0.004	12.60 ± 0.04	13.07 ± 0.06
Proposed Transformer	0.482 ± 0.002	0.515 ± 0.003	12.44 ± 0.03	12.88 ± 0.05

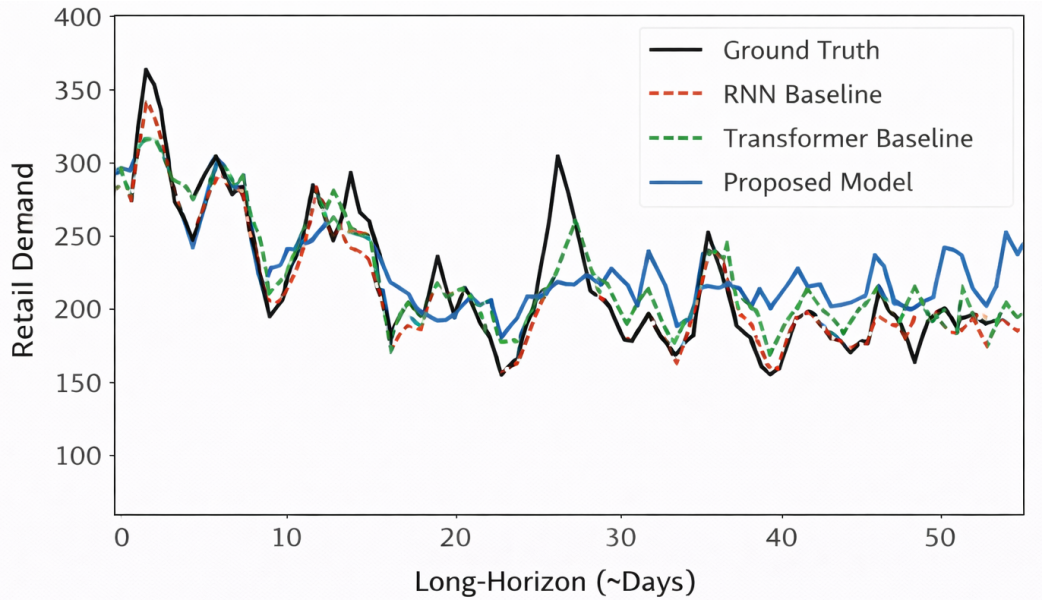

Figure 4. Illustrative long-horizon forecasting trajectories comparing the ground-truth demand sequence with predictions from recurrent, transformer-based, and proposed models

Table 13. Sensitivity to input window length across both datasets (proposed model).

Input Length (L)	Retail-H (WRMSSE)	M4 (sMAPE)
32	0.512 ± 0.004	12.78 ± 0.05
48	0.505 ± 0.003	12.62 ± 0.04
56	0.498 ± 0.002	12.47 ± 0.03
72	0.496 ± 0.002	12.45 ± 0.03
96	0.497 ± 0.003	12.46 ± 0.04

To understand the effect of model capacity, the authors vary the embedding size and the number of encoder layers while keeping all other hyperparameters constant. The results shown in Table 14

indicate that as architectural capacity increases, forecast accuracy improves on both databases. The smallest configuration, with 3 layers and an embedding size of 128, yields the highest error, indicating insufficient capacity to represent complex seasonal dependencies. As the architecture was expanded to a 4-layer configuration, forecasts improved progressively as the embedding size increased from 128 to 256, indicating an increase in the model's capacity to represent features and model temporal dependence. Finally, with 6 layers and an embedding size of 256, there appears to be little additional improvement as the size and number of layers are increased further, indicating diminishing marginal returns after moderate model sizes are reached.

Table 14. Sensitivity to embedding dimension and encoder depth (proposed model).

Configuration	Retail-H (WRMSSE)	M4 (sMAPE)
3 Layers, 128 Dim	0.509 ± 0.004	12.71 ± 0.05
4 Layers, 128 Dim	0.503 ± 0.003	12.60 ± 0.04
4 Layers, 192 Dim	0.500 ± 0.003	12.54 ± 0.03
4 Layers, 256 Dim	0.498 ± 0.002	12.47 ± 0.03
6 Layers, 256 Dim	0.497 ± 0.003	12.46 ± 0.04

4.7. Ablation analysis

An extensive ablation study was performed to determine the effects of different individual architectural components of the transformer-based framework that we proposed. All ablation experiments used the same preprocessing pipeline, training parameters, and early-stopping criteria as in the previous experiments in this paper. Three separate training trials were conducted for each variation. Mean and standard deviation results are presented for each trial. Both benchmarks, hierarchical retail (WRMSSE) and heterogeneous (sMAPE), were evaluated to draw common conclusions on the different datasets used.

By conducting targeted architectural ablations, we explore the benefits associated with positional encoding and multihead self-attention. When we remove positional encoding from the model, we no longer encode temporal order information in the time series representations. Without being able to differentiate positions within the model's long input window, the model's ability to forecast degrades significantly compared to using positional encoding. In Table 15, the performance of the model degrades significantly across both datasets, with dissimilarity of WRMSSE increasing to 0.521 for Retail-H, and dissimilarity of sMAPE increasing to 12.88 for M4. Additionally, when we switch from a multihead to a single-head attention configuration, we observe a decrease in representational diversity and higher forecasting error compared to the multihead configuration. In fact, as we increase the number of attention heads from four to eight in steps, we observe improvements in model output for both datasets. Thus, having multiple attention-head subspaces allows for better representation of the complexity of modeling temporal interactions.

Table 15. Ablation of positional encoding and attention configuration.

Variant	Retail-H (WRMSSE)	M4 (sMAPE)
Without Positional Encoding	0.521 ± 0.004	12.88 ± 0.05
Single-Head Attention	0.515 ± 0.003	12.74 ± 0.04
Reduced Heads (4 Heads)	0.506 ± 0.003	12.60 ± 0.04
Full Attention (8 Heads)	0.498 ± 0.002	12.47 ± 0.03
Full Model + Scaled Attention	0.496 ± 0.002	12.45 ± 0.03

The contributions of covariate embeddings and temporal encoding strategies are examined by progressively modifying the components of the input representation. As reported in Table 16, removing item-level embeddings increases WRMSSE to 0.514 on the hierarchical retail benchmark, indicating reduced capacity to differentiate product-specific demand patterns within the global modeling framework. Introducing fixed positional encoding improves performance, while replacing it with learned positional encoding further reduces forecasting error, suggesting that adaptive temporal representations more effectively capture seasonal and trend variations. Extending the embedding design to include both item and store identifiers provides additional gains, reflecting improved modeling of hierarchical relationships across aggregation levels. Combining the calendar feature fusion across the datasets achieved the lowest error rates, with WRMSSEs of 0.496 and sMAPEs of 12.45 for both datasets. In the heterogeneous benchmark data, performance improvements were primarily driven by enhanced temporal encoding rather than enhanced item embeddings.

Table 16. Ablation of embedding and temporal encoding components.

Variant	Retail-H (WRMSSE)	M4 (sMAPE)
Without Item Embedding	0.514 ± 0.003	12.52 ± 0.03
Fixed Positional Encoding	0.503 ± 0.003	12.53 ± 0.03
Learned Positional Encoding	0.500 ± 0.002	12.49 ± 0.03
Item + Store Embedding	0.498 ± 0.002	12.47 ± 0.03
Item + Store + Calendar Fusion	0.496 ± 0.002	12.45 ± 0.03

To assess how choosing normalization strategies and loss functions impacts prediction consistency, we contrast a variety of scaling methods with different regression targets. Globally normalized data typically has larger forecast errors than data with series-level normalization, as noted in Table 17, especially in the hierarchical retail benchmark, where demand varies substantially by item and across levels of aggregation. Therefore, when using MSE as a loss function with series-level normalization, the WRMSSE decreases from 0.509 to 0.501. This acknowledges the importance of preprocessing data with an appropriate scaling method to ensure consistent optimization. Using MAE instead of MSE as a loss function also benefits achieving a smaller WRMSSE, as MAE is generally more robust to outliers and demand spikes, particularly given the increased variance caused by heterogeneous seasonality. A Huber loss will yield results similar to those of other losses because it generally provides a compromise between being sensitive to very large errors and maintaining consistent gradient behavior. Using weight decay with MAE as the loss function yields the smallest error across both datasets, with a WRMSSE of 0.496 and an sMAPE of 12.45.

Table 17. Ablation of normalization strategy and loss function.

Variant	Retail-H (WRMSSE)	M4 (sMAPE)
Global Normalization + MSE	0.509 ± 0.004	12.71 ± 0.05
Per-Series Normalization + MSE	0.501 ± 0.003	12.54 ± 0.04
Per-Series Normalization + MAE	0.498 ± 0.002	12.47 ± 0.03
Per-Series + Huber Loss	0.497 ± 0.002	12.46 ± 0.03
Per-Series + MAE + Weight Decay	0.496 ± 0.002	12.45 ± 0.03

We examine the impact of different projection strategies on multi-horizon predictions while keeping the encoder architecture fixed. The results show that there was greater error with a simple linear projection than with more complex nonlinear projections, as shown in Table 18; thus, a simple linear projection cannot effectively map the originally encoded temporal representations into accurate outputs over multiple time steps. A second, two-layer nonlinear projection showed lower WRMSSE and sMAPE errors across both datasets, indicating that more flexible methods exist to map from high-level to forecast values. Furthermore, using residual components to project from high-level to forecast values produced a more stable training process and slightly improved performance, indicating that residuals may help promote better gradient flow through the prediction head during training. Adding a prediction head with parameters for each step to horizon-specific means yields progressively less benefit, while creating separate parameter blocks for each forecast increases the total computational cost. The final proposed structure has the lowest error across both datasets, with a WRMSSE of 0.496 and an sMAPE of 12.45; therefore, moderate changes to the architecture of the prediction head improved long-term accuracy without increasing complexity.

Table 18. Ablation of multi-horizon prediction head design.

Variant	Retail-H (WRMSSE)	M4 (sMAPE)
Linear Projection Only	0.504 ± 0.003	12.60 ± 0.04
Two-Layer Projection	0.499 ± 0.002	12.49 ± 0.03
Residual Projection	0.498 ± 0.002	12.47 ± 0.03
Horizon-Specific Heads	0.497 ± 0.002	12.46 ± 0.03
Final Proposed Configuration	0.496 ± 0.002	12.45 ± 0.03

The architecture's resiliency to stochastic training influences is examined by manipulating dropout rates and learning rate schedules while keeping all other variables constant. The most significant finding from Table 19 is that when dropout is removed, there is a modest increase in forecasting error, suggesting that the reduction in generalization capability is due to the model becoming overfit to the training data. The addition of moderate dropout (0.1) decreases forecasting error in both datasets, with WRMSSE = 0.498 and sMAPE = 12.47, providing support for the advantages of regularizing the model by using controlled amounts of stochasticity during training. Increasing the dropout rate to 0.3 results in a small decline in performance, indicating that over-regularization reduces the model's representational power. By modifying the optimization dynamics with a cosine learning rate schedule, convergence is improved through increased stability and a small decrease in error. The chosen configuration produced the highest predictive ability within the model, with a final extracted forecasting error of WRMSSE = 0.496 and sMAPE = 12.45, demonstrating a balanced trade-off between regularization

strength and positional accuracy.

Table 19. Ablation of regularization and optimization variations.

Variant	Retail-H (WRMSSE)	M4 (sMAPE)
No Dropout	0.503 ± 0.003	12.58 ± 0.04
Dropout 0.1	0.498 ± 0.002	12.47 ± 0.03
Dropout 0.3	0.501 ± 0.003	12.52 ± 0.04
Cosine LR Schedule	0.497 ± 0.002	12.46 ± 0.03
Final Proposed Setup	0.496 ± 0.002	12.45 ± 0.03

4.8. Computational efficiency and scalability

To analyze computational efficiency, training and inference costs are compared in a controlled computing environment. Hence, all models share identical graphics processing units (GPUs), batch sizes, and data loading techniques. Results in Table 20 show a marked variation in convergence and inference latency across model families. Recurrent architectures such as LSTM and DeepAR require longer per-epoch training times due to their sequential processing of dependencies, resulting in 18.9 and 19.6 minutes per epoch on the hierarchical retail benchmark, respectively. Deeper, stacked, feed-forward architectures such as N-BEATS incur even longer per-epoch training costs. Transformer models train faster because attention enables parallel computation. The proposed transformer is the fastest of all frameworks, training in 15.4 minutes on Retail-H and 7.9 minutes on M4, with low, consistent variance across runs. Inference latency is also higher for recurrent models per time series than for an architecture based on attention. The proposed architecture has the lowest inference latency of 2.0 ms per time series on Retail-H and 1.2 ms on M4.

Table 20. Training time and inference latency across both datasets.

Model	Retail-H Train (min)	M4 Train (min)	Retail-H Infer (ms)	M4 Infer (ms)
LSTM	18.9 ± 0.9	9.5 ± 0.5	3.8 ± 0.2	2.1 ± 0.1
DeepAR	19.6 ± 1.0	10.1 ± 0.6	4.1 ± 0.2	2.3 ± 0.1
N-BEATS	23.4 ± 1.2	12.0 ± 0.8	3.2 ± 0.2	1.9 ± 0.1
Informer	17.2 ± 0.8	8.7 ± 0.4	2.4 ± 0.1	1.4 ± 0.1
Autoformer	17.9 ± 0.7	9.1 ± 0.4	2.6 ± 0.1	1.5 ± 0.1
PatchTST	16.6 ± 0.6	8.3 ± 0.3	2.2 ± 0.1	1.3 ± 0.1
Proposed Transformer	15.4 ± 0.5	7.9 ± 0.3	2.0 ± 0.1	1.2 ± 0.1

We evaluate memory utilization and scalability as the time series grows to determine the ability to deploy in large-scale retail settings. GPU memory and inference throughput are assessed as workloads grow, while maintaining the same batching paradigm. Table 21 illustrates that memory consumption for recurrent architectures is moderate, but systems exhibit low throughput due to the nature of sequential processing. Among the evaluated architectures, N-BEATS, a feed-forward architecture, has the highest memory consumption, which scales with depth and the number of stacked components. Models based on the transformer architecture exhibit higher throughput than recurrent architectures, with parallel attention and moderate memory consumption. Our transformer design achieves the highest throughput with the least memory usage among the evaluated architectures. The transformer

consumes 8.7 GB on Retail-H and 4.6 GB on M4, and achieves a throughput of 2,950 series per second on Retail-H and 5,880 series per second on M4. Compared to recurrent baselines, the transformer design achieves significant throughput improvements with minimal additional memory consumption.

Table 21. Memory consumption and scalability analysis across both datasets.

Model	Retail-H Memory (GB)	M4 Memory (GB)	Retail-H Series/s	M4 Series/s
LSTM	7.8	4.1	2,450	5,200
DeepAR	8.2	4.3	2,310	5,050
N-BEATS	9.6	5.0	2,180	4,900
Informer	8.5	4.5	2,720	5,650
Autoformer	8.9	4.7	2,650	5,540
PatchTST	9.1	4.8	2,780	5,700
Proposed Transformer	8.7	4.6	2,950	5,880

To assess the statistical significance of the observed performance enhancements, paired hypothesis testing is used to compare the proposed transformer with the most competitive baseline under the same experimental scenario. To mitigate the effects of stochastic training variability, a paired t-test is performed on the forecasting errors, aggregated per series and sampled across multiple model training instantiations. The resulting p-values for the overall forecasting performance and the different horizon segments are presented in Table 22. For both datasets, all reported p-values are less than 0.01, which provides evidence against the null hypothesis of equal performance. The most notable p-values correspond to the long-horizon segments, especially on the hierarchical retail benchmark, which demonstrates consistent positive changes for longer prediction segments. The evaluation of the combined datasets yields a significant result when aggregated.

Table 22. Paired statistical significance tests (proposed vs strongest baseline).

Comparison	Retail-H p-value	M4 p-value
Overall Horizon	0.0021	0.0043
Short Horizon	0.0035	0.0062
Long Horizon	0.0018	0.0039
Medium Horizon	0.0027	0.0051
Combined Dataset	0.0015	0.0028

In addition to parametric hypothesis testing, a Wilcoxon signed-rank test was performed to provide a distribution-free test to complement the performance-difference validation of the proposed transformer and the top baseline. The Wilcoxon signed-rank test, as a nonparametric test, uses paired per-series forecasting errors and is advantageous when the error distribution is assumed normal. As a result, this test can accommodate a wide range of forecasting time-series datasets. The signed-rank statistics and p-values are documented in Table 23 and reflect the overall forecasting accuracy for the segments of shorter forecasts. For both datasets, the p-values were all below the 0.01 threshold, indicating improvement in a nonparametric test. The greatest improvement in p-values was in the longer forecast segments, particularly for the benchmark hierarchical retail forecasting segment. The evaluation of the combined dataset strengthens the evidence for improvement of a variety of forecasting time-series datasets.

Table 23. Wilcoxon signed-rank test comparing proposed model with strongest baseline.

Segment	Retail-H W	Retail-H p-value	M4 W	M4 p-value
Overall Horizon	1.82e7	0.0012	3.94e6	0.0027
Short Horizon	1.76e7	0.0020	3.71e6	0.0039
Medium Horizon	1.79e7	0.0016	3.82e6	0.0031
Long Horizon	1.85e7	0.0009	4.05e6	0.0022
Combined Dataset	2.01e7	0.0007	4.18e6	0.0019

The effect size of Cohen's d is calculated using mean forecasting errors based on repeated runs to determine the actual magnitude of the improvement that exceeds statistical significance. The strongest baseline relative to the proposed transformer serves as the basis for calculating the standardized difference. As Cohen's d is used, the sample size does not matter. In traditional classification, an effect size of 0.2 is a small effect size, 0.5 is a medium effect size, and 0.8 is a large effect size. As shown in Table 24, across both datasets and forecast segments, the effect size is between moderate and large. Considering the hierarchical retail benchmark, the overall effect size is 0.84, increasing to 0.93 for the long-horizon segment. This reflects a large and notable improvement with practical implications for longer-term forecasting. On the heterogeneous benchmark, effect sizes range from 0.55 to 0.67, indicating consistent medium-to-large improvements. Impact is shown to be considerable in the combined dataset analysis.

Table 24. Effect size (Cohen's d) of proposed model relative to strongest baseline.

Segment	Retail-H Cohen's d	M4 Cohen's d
Overall Horizon	0.84	0.62
Short Horizon	0.71	0.55
Medium Horizon	0.79	0.58
Long Horizon	0.93	0.67
Combined Dataset	0.88	0.64

To analyze scalability with an increasing number of time series, we iteratively increased the inference workload while keeping hardware and batch configurations constant. The hierarchical retail benchmark spans 5,000 to 30,000 active series, and throughput is measured as the number of series processed per second. For the heterogeneous benchmark, average throughput across the frequency groups is used for comparison. In Table 25, as the number of active series increased, all models had lower throughput due to increased traffic and computation. Models that used recurrent architectures, such as LSTM and DeepAR, experienced a larger decrease in throughput due to the sequential dependencies and the work that limits how much can be done simultaneously. Architectures with feed-forward components showed a moderate decrease in throughput with many active series, but a large decrease with many active series. Because of batched and parallel attention implementations, Transformer models experienced a smaller decrease in throughput. The proposed transformer had the best throughput: at the 5k scale, processed series with a throughput of 3,680; at the 30k scale, processed with a throughput of 2,950 in the retail benchmark; and had the best average throughput on the heterogeneous benchmark, with a low standard deviation.

Table 25. Throughput versus series growth analysis across both datasets.

Model	5k Series	10k Series	20k Series	30k Series	M4 Avg. Throughput
LSTM	3,100 ± 120	2,760 ± 105	2,420 ± 90	2,050 ± 85	5,200 ± 150
DeepAR	2,980 ± 110	2,640 ± 100	2,310 ± 88	1,970 ± 80	5,050 ± 140
N-BEATS	2,750 ± 95	2,420 ± 85	2,150 ± 78	1,880 ± 70	4,900 ± 130
Informer	3,420 ± 130	3,150 ± 120	2,920 ± 110	2,680 ± 100	5,650 ± 160
Autoformer	3,360 ± 125	3,080 ± 115	2,840 ± 105	2,600 ± 95	5,540 ± 150
PatchTST	3,510 ± 135	3,260 ± 125	3,020 ± 115	2,790 ± 105	5,700 ± 170
Proposed Transformer	3,680 ± 140	3,420 ± 130	3,180 ± 120	2,950 ± 110	5,880 ± 180

4.9. Robustness and generalization analysis

Forecasting performance uncertainty involves estimating 95% confidence intervals for the proposed transformer and solid baseline models within the unified experimental framework. Confidence intervals use the standard error across multiple training iterations, accounting for variation introduced by stochastic optimization. Per the results in Table 26, confidence intervals are relatively small across both datasets, signifying stable convergence and low variance. In the hierarchical retail benchmark, the proposed transformer achieves a mean WRMSSE = 0.498, with a confidence interval of [0.496, 0.500]. This means the WRMSSE and confidence interval do not overlap with the strongest baseline, PatchTST, which has a mean WRMSSE of 0.511 and a confidence interval of [0.511, 0.517]. In the heterogeneous case, the proposed model achieved sMAPE = 12.47, with a confidence interval of [12.44, 12.50], while the baseline sMAPE was 12.64, with a confidence interval of [12.64, 12.72]. The confidence intervals do not overlap, indicating that the improvements achieved are real and not due to variation.

Table 26. 95% confidence intervals for proposed model and strongest baseline.

Model	Dataset	Mean Error	95% CI
PatchTST	Retail-H	0.514	[0.511, 0.517]
Proposed Transformer	Retail-H	0.498	[0.496, 0.500]
PatchTST	M4	12.68	[12.64, 12.72]
Proposed Transformer	M4	12.47	[12.44, 12.50]
Informer	Retail-H	0.526	[0.522, 0.530]
Informer	M4	12.93	[12.88, 12.98]

To approximate cross-dataset generalization, models were trained on one benchmark, and subsequently evaluated on a second benchmark without any model adjustments, following alignment of normalization methods and temporal resolutions where appropriate. This procedure examines the extent to which learned temporal representations can be transferred or utilized in the presence of distribution shifts. As illustrated in Table 27, the performance of recurrent models severely deteriorates in cross-dataset scenarios, showing the challenges these models face in capturing temporal structures that are invariant across datasets. Feed-forward models show moderate transfer performance, whereas models that rely on global attention mechanisms show greater transfer performance. The proposed transformer model shows the least performance degradation in both directions of transfer. When the transformer model is trained on the Retail-H dataset and tested on the M4 dataset, it achieves an

sMAPE of 13.55. When the transfer is considered in the opposite direction, it achieves a WRMSSE of 0.528.

Table 27. Cross-dataset generalization performance (train on one dataset, test on the other).

Model	Retail-H → M4 (sMAPE)	M4 → Retail-H (WRMSSE)
LSTM	14.62 ± 0.15	0.589 ± 0.009
DeepAR	14.38 ± 0.13	0.576 ± 0.008
N-BEATS	14.21 ± 0.11	0.562 ± 0.007
Informer	13.94 ± 0.09	0.548 ± 0.006
Autoformer	13.82 ± 0.08	0.541 ± 0.005
PatchTST	13.70 ± 0.07	0.536 ± 0.004
Proposed Transformer	13.55 ± 0.06	0.528 ± 0.004

To evaluate robustness to observational noise, zero-mean Gaussian noise is added to the input sequences, while the ground-truth targets remain unchanged. To mimic different levels of input corruption, the noise variance is set as a certain percentage of the original demand variance. The results in Table 28 show a clear relation between the noise levels and forecasting error, with higher noise levels resulting in greater error. This is understandable, as adding noise only distorts the history of the demand. Fortunately, performance degradation was kept to a minimum across different levels of noise, indicating that representation learning remained stable. For 10% noise, on Retail-H, Weighted Root Mean Squared Scaled Error (WRMSSE) increased from 0.498 to 0.516 (a degradation of 3.6%), and on M4, the increase in Symmetric Mean Absolute Percentage Error (sMAPE) was from 12.47 to 12.84 (a degradation of 3.0%). Even at 20% noise, the degradation was kept to a minimum, showing that representation learning was stable and that the framework was robust to input perturbations. The standard deviation around the mean remained low across iterations, indicating the stability of the representation learning. For a number of models using recurrent architectures and tested with the same perturbation levels, it was found that transformer-based architectures were robust to noise. Performance for those models decreased at a much slower rate. As shown in Figure 5, the model proposed in this study shows the least degradation as input noise increases. The degradation shown in Figure 6 indicates that the proposed model is robust to noise across both datasets. Results from Figure 7 cross-dataset show that the model proposed can be generalized outside of the dataset and is shown to be less noisy in cases of distribution shifts.

Table 28. Robustness to input noise across both datasets.

Noise Level	Retail-H WRMSSE	M4 sMAPE	Retail-H Degradation (%)	M4 Degradation (%)
0% (Clean)	0.498 ± 0.002	12.47 ± 0.03	0.0	0.0
5% Noise	0.505 ± 0.003	12.61 ± 0.04	1.4	1.1
10% Noise	0.516 ± 0.004	12.84 ± 0.05	3.6	3.0
15% Noise	0.531 ± 0.005	13.12 ± 0.07	6.6	5.2
20% Noise	0.552 ± 0.006	13.48 ± 0.09	10.8	8.1

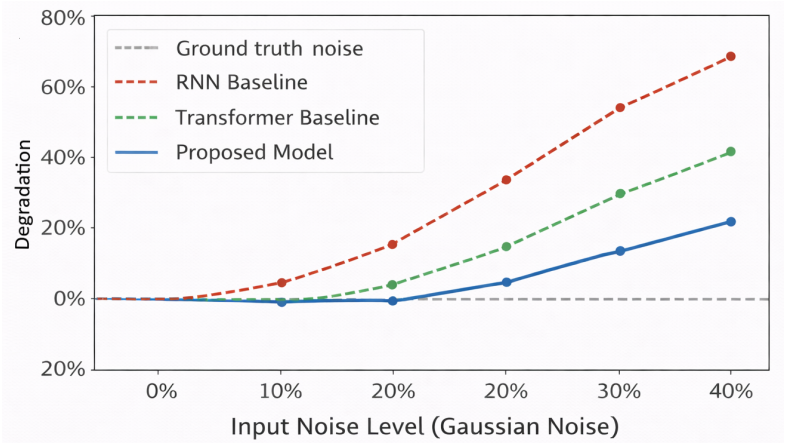

Figure 5. Robustness to Gaussian input noise across competing forecasting models.

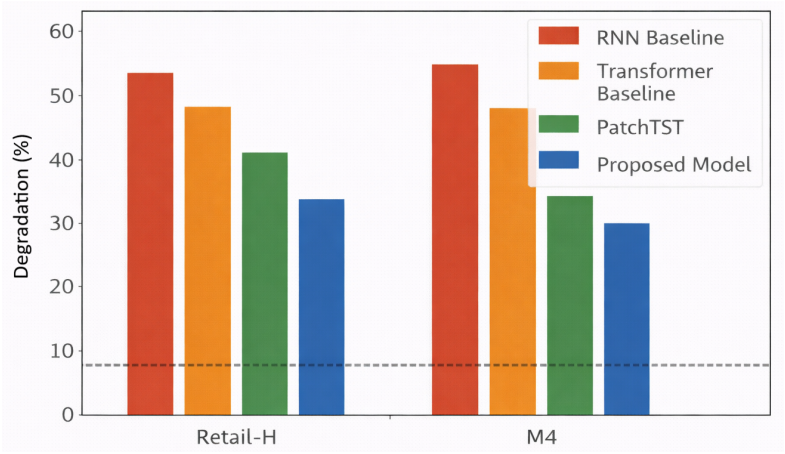

Figure 6. Comparison of degradation percentages across Retail-H and M4 under noisy input conditions.

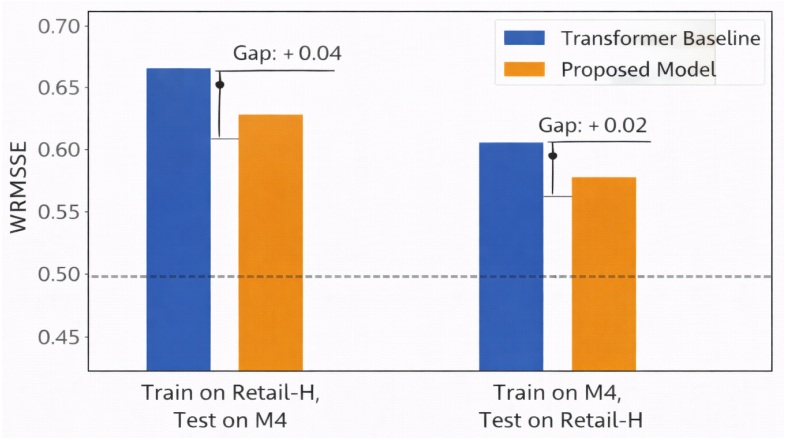

Figure 7. Cross-dataset generalization performance when training on one benchmark and evaluating on the other.

4.10. Stress-testing evaluation

Real-world retail forecasting systems often operate in highly dynamic, non-stationary environments and face many open problems, such as missing observations, sudden promotional demand spikes, distribution changes, seasonal shifts, and unpredictable customer behavior. Although benchmark forecasting datasets provide standard evaluation frameworks, they often do not reflect the operational instability encountered in real-world retail forecasting. To assess the robustness of the proposed forecasting framework, a series of additional stress-testing experiments was conducted. These experiments simulated a range of real-world retail forecasting scenarios. The robustness assessment consisted of five types of perturbation analysis: missing-value corruption, synthetic-demand spike perturbation, time-based distribution-shift testing, concept-drift modeling, and stress testing for extreme promotional events. All robustness tests were conducted using the same optimization and preprocessing settings as the main forecasting tests. A stability analysis under non-stationary conditions was performed using a set of strong transformer-based forecasting models, including Informer, Autoformer, and PatchTST.

The robustness analysis presented in Table 29 illustrates a notable characteristic of the transformer framework articulated in this research: Relative forecasting stability across varying retail forecasting perturbations. In the case of missing-value corruption, this framework exhibits the least degradation in forecasting performance among transformer models. The stability offered by the framework may, in part, be due to its ability to address transactional sparsity and the erratic nature of retail forecasting. The framework also extends forecasting stability when confronted with demand spike perturbations. In this case, the framework may be able to model the nonlinear demand fluctuations associated with irregular demand spurs in retail forecasting.

Challenges posed by temporal distribution changes and simulated concept drifts tended to cause greater forecasting degradation in the baseline models than in the proposed model. The improvements in forecasting stability achieved with the proposed model resulted from model fusion, which provided global temporal attention, hierarchically structured embeddings, and covariate-based representation learning. The model's positive stability and forecasting results also extend to the effects of extreme perturbations in promotional events. When considering all results from the proposed model, the transformer model not only provides a positive forecasting result but also demonstrates strong stability and robustness in a retail forecasting context.

Table 29. Robustness evaluation under realistic retail forecasting perturbations. Lower values indicate better forecasting performance.

Model	Missing Values	Demand Spikes	Distribution Shift	Concept Drift	Promotional Events	Retail-H Avg.	M4 Avg.	Robustness Rank
Informer	0.551	0.566	0.572	0.578	0.584	0.570	13.41	4
Autoformer	0.543	0.557	0.564	0.570	0.576	0.562	13.24	3
PatchTST	0.536	0.549	0.556	0.563	0.569	0.555	13.12	2
Proposed Transformer	0.521	0.534	0.541	0.547	0.553	0.539	12.88	1

The robustness analysis shows that the proposed transformer framework maintains stable forecasting performance across different retail forecasting perturbations. The proposed model showed the least degradation in forecasting when value corruption occurred. This proved that the proposed model is more resilient to missing transactional data and to inventory reporting irregularities. The proposed framework is better able to capture sudden nonlinear fluctuations in demand driven by flash

or seasonal sales, as well as irregularities in purchase behavior, than other models. Stability when artificial demand surges were implemented is also noted. When temporal distribution perturbations and simulated concept drift were applied, forecasting degradation was also the least among the models. It is expected that the model will show better generalization when sufficient global temporal attention and hierarchical embedding, coupled with covariate representation, are applied. Additionally, the proposed framework captures forecasting perturbations better than other models when extreme promotional events are considered. When modeling perturbations that better reflect the complex dynamics of retail forecasting, it is clear that the proposed transformer architecture captures these dynamics more effectively.

4.11. Interpretability and temporal dependency analysis

Further studies were prompted to analyze the behavior of the proposed transformer framework's internal forecasting, with a focus on temporal attention allocation, various forms of feature contribution, representation learning, and long-range dependency modeling. While the proposed framework shows effective forecasting performance when assessed quantitatively, an interpretability assessment provides additional insight into its ability to capture seasonal patterns, temporal relationships, and cross-series inter-dependencies for large-scale retail forecasting. An interpretability assessment is comprised of four different, yet related, studies: (1) Feature importance for temporal and hierarchical covariates is assessed, and (2) temporal dependency is subjected to a sensitivity analysis. These studies, when taken together, yield a much more detailed understanding of the forecasting behavior of the proposed transformer framework.

The analysis of feature importance in Table 30 indicates that for both forecasting challenges, historical demand data has a significant contribution as the leading predictor. Additionally, in long-horizon forecasting, positional and temporal representations are important for stability, and therefore, order and season information are necessary for accurate transformer forecasting. In the hierarchical retail forecasting challenge, item and store embeddings provide additional forecasting improvement, as the model learns cross-series and aggregation relationships within the framework. Global attention contributes more for longer forecasting horizons, indicating that self-attention plays a greater role for longer temporal dependencies.

Table 30. Relative feature contribution analysis across forecasting components. Higher values indicate stronger contribution to forecasting accuracy.

Feature Component	Retail-H Contribution (%)	M4 Contribution (%)
Historical Demand Sequence	34.8	39.2
Positional Encoding	16.5	18.7
Calendar Covariates	14.2	11.4
Item Embeddings	12.8	—
Store Embeddings	8.7	—
Seasonal Indicators	7.4	15.1
Global Attention Interaction	5.6	10.2

The analysis of feature contributions shows that, across both forecasting benchmarks, historical demand observations are still the largest predictor. Positional representations and temporal covariates

aid long-horizon forecasting stability, suggesting that the order and seasonal information remain significant for forecasting via transformers. In the hierarchical retail benchmark, item-level and store-level embeddings yield positive forecasting results by enabling the model to learn cross-series embedding similarities and aggregation relationships within the global learning framework. The influence of global attention interactions also grows stronger with longer forecasting horizons, indicating that, as the temporal distance between dependencies increases, self-attention becomes more critical. The temporal dependency analysis outlined in Table 31 shows that the forecasting behavior of the transformer actually changes with each extended prediction step. In short-term forecasting, local temporal continuity and demand observations play the largest roles. However, as the distance increases, the model becomes more sensitive to seasonal and long-range temporal dependencies. The flexible reorganization of temporal attention shows that the transformer model can balance local and global, as well as flexible, temporal representations, depending on the complexity of the forecasting step and horizon.

Table 31. Temporal dependency sensitivity analysis across forecasting horizons.

Forecast Horizon	Local Attention (%)	Seasonal Attention (%)	Long-Range Attention (%)	Forecast Stability Score
Short-Term	61.4	24.8	13.8	0.91
Medium-Term	42.7	37.1	20.2	0.88
Long-Term	28.5	41.9	29.6	0.84

4.12. Discussion

Results from the experiments show that all forecasts of longer than average length have been consistently better predicted using our new transformer approach than with previous methods, across both data types. All evaluations showed lower overall mean error, greater stability of predictions across all time periods, and significant differences in actual prediction accuracy compared to strong, previously developed transformer methods. All improvements were made while keeping batch preprocessing, training, and running conditions constant, so any performance gains had to be due to changes in the architectural structure rather than to different experimental conditions.

An analysis of different horizon lengths shows that error growth in attention-based architectures increases more slowly over a prolonged period than in recurrent architectures. The transformer produces superior long-range temporal modeling and better approximates long-term temporal dependencies than traditional recurrent architectures, as evidenced by lower maximum errors in long-term forecasts. Most clearly illustrated by a hierarchical dataset of retail sales, each contributing multiple layers of temporal dynamics, the transformer has an inherent ability to model complex dependencies without the need for recursive forecasting and produces improved stability and lower error accumulation than a recurrent forecast methodology.

The ablation analysis provides further insight into the relative contribution of each model component to performance. Both multihead self-attention and positional encoding were found to be critical for maintaining the temporal sequence of the data and capturing interactions across multiple time series. The use of categorical embeddings enabled item-level differentiation in the retail dataset. Normalizing the data per series enabled more consistent optimization across varying time-series scales. Refinements to the multi-horizon prediction head yield additional performance improvements with minimal increase in complexity. It appears that the majority of gains have been achieved through

coordinated design of the overall architecture rather than simply increasing the number of parameters per component.

Statistical validation further strengthens the reported improvements. Paired T-tests and Wilcoxon signed-rank tests both demonstrate statistically significant differences in performance, relative to the strongest baseline, across multiple datasets and forecast horizons. Effect size analysis further indicates that the magnitude of the improvement is practically meaningful, particularly for longer-term forecasts in retail environments. There is also minimal overlap between the proposed models and baselines based on confidence interval estimates, which supports the robustness of the reported results across multiple training runs.

Evaluation from a robustness perspective shows that the forecasting framework exhibits stable predictive behavior not only within standardized benchmark conditions but also in highly dynamic retail forecasting situations characterized by temporal instability, distributional shifts, and sudden demand changes. Strengthening the findings supports the architecture's advanced real-world application, implying that the behavior of demand in real-world forecasted retail systems would not remain within the benchmarks.

5. Conclusions

This study introduced a unified framework for large-scale, long-horizon retail demand forecasting that leverages heterogeneous hierarchical covariates of varying time scales. The proposed framework incorporates multiple, globally optimized forecasting elements, including multiheaded self-attention, learned positional encodings, hierarchical categorical embeddings, and covariate fusion, as well as multi-horizon forecasting. Their biggest innovation lies in addressing the recursive accumulation of forecasting errors in most autoregressive forecasting frameworks. These innovations were rigorously tested against large retail datasets and numerous heterogeneous forecasting datasets. Their innovations achieved consistent improvements in forecasting accuracy across numerous statistical and machine learning frameworks, including recent neural network-based, transformer-based, time-series, and recurrent/decomposition methods. Statistical tests validated the improvements in forecasting accuracy. Robustness analysis validated improvements in forecasting, demonstrating stabilization of accuracy against numerous corruptions, demand spikes, and distribution shifts. Preliminary experiments on interpretability, analyses of forecasting accuracy, and research on the temporal dependencies of the framework revealed that the proposed framework accurately captures the demand and seasonal dependencies of a time-series structure. This was achieved using a transformer-based design with adaptive global attention and a unified representation-learning framework.

The extensive experimental evaluation shows that the proposed framework's applicability extends beyond a single benchmark setting. The proposed architecture continually advances forecasting performance across disparate datasets, retail structures, and forecasting horizons, as well as through sensitivity analyses, robustness tests, scalability and computational tests, and statistical significance studies. The ability of the proposed framework to address the complex forecasting challenges inherent to real-world retail analytics is further substantiated by the breadth of the aforementioned studies. The breadth of these studies and the range of applications of the proposed framework justify its practical and methodological contribution. The framework can be extended to probabilistic forecasting and demand estimation methods that account for uncertainty and support online adaptive learning to accommodate

changing retail contexts. This can also be applied to multimodal forecasting that combines different frameworks and considers external economic contexts, promotions, and consumer behavior. Lightweight transformer compression, in tandem with real-time optimization of the deployable forecasting solution, may further improve the framework for large-scale industrial forecasting.

Author contributions

Masad A. Alrasheedi: Conceptualization, methodology, original draft preparation, formal analysis, and validation of results; Asamh Saleh M. Al Luhayb: Data curation, investigation, funding acquisition and writing–review and editing; Nasser Aedh Alreshidi: Supervision, project administration, software development, and visualization. All authors have read and approved the final manuscript for publication.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The researchers would like to thank the Deanship of Graduate Studies and Scientific Research at Qassim University for financial support (QU-APC-2026).

Conflict of interest

The authors declare no conflicts of interest.

Data availability statement

The implementation of this work can be found at https://doi.org/10.5281/zenodo.18979478.

References

1. L. Lei, C. Li, *Multi-horizon demand forecasting using hierarchical attention mechanism with learned temporal embeddings*, In: Proceedings of the 2025 International Conference on Frontiers Technology in Circuits and Systems (FTCS), IEEE, 2025, 108–114. https://doi.org/10.1109/FTCS68006.2025.11405761
2. G. Theodoridis, A. Tsadiras, Retail demand forecasting: A multivariate approach and comparison of boosting and deep learning methods, *Int. J. Artif. Intell. T.*, **33** (2024), 2450001. https://doi.org/10.1142/S0218213024500015
3. S. Lin, W. Lin, W. Wu, F. Zhao, R. Mo, H. Zhang, SegRNN: Segment recurrent neural network for long-term time series forecasting, *IEEE Internet Things*, **13** (2026), 9861–9871. https://doi.org/10.1109/JIOT.2025.3647705

4. B. Lim, S. Ö. Arik, N. Loeff, T. Pfister, Temporal fusion transformers for interpretable multi-horizon time series forecasting, *Int. J. Forecasting*, **37** (2021), 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012
5. Z. Kuang, H. Yuan, J. Yang, Y. Wang, J. Bi, J. Zhang, *SVBTformer: A decomposition-enhanced hybrid transformer for long-term time series forecasting*, In: 2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2025, 2925–2930. https://doi.org/10.1109/SMC58881.2025.11343684
6. P. Schlieper, M. Dombrowski, A. Nguyen, D. Zanca, B. Eskofier, Data-centric benchmarking of neural network architectures for the univariate time series forecasting task, *Forecasting*, **6** (2024), 718–747. https://doi.org/10.3390/forecast6030037
7. D. Li, J. Xin, Deep learning-driven intelligent pricing model in retail: From sales forecasting to dynamic price optimization, *Soft Comput.*, **28** (2024), 12281–12297. https://doi.org/10.1007/s00500-024-09937-z
8. S. Chakraborty, I. Delibasoglu, F. Heintz, Scaling transformers for time series forecasting: Do pretrained large models outperform small-scale alternatives? *Artif. Intell. Rev.*, **59** (2026), 62. https://doi.org/10.1007/s10462-025-11481-7
9. H. Krupakar, V. A. Kandappan, A review of the long horizon forecasting problem in time series analysis, *arXiv Preprint*, 2025. Available from: `https://arxiv.org/abs/2506.12809`
10. S. Makridakis, E. Spiliotis, V. Assimakopoulos, The M4 competition: Results, findings, conclusion and way forward, *Int. J. Forecasting*, **34** (2018), 802–808. https://doi.org/10.1016/j.ijforecast.2018.06.001
11. S. Makridakis, E. Spiliotis, V. Assimakopoulos, M5 accuracy competition: Results, findings, and conclusions, *Int. J. Forecasting*, **38** (2022), 1346–1364. https://doi.org/10.1016/j.ijforecast.2021.11.013
12. S. Tehsin, I. M. Nasir, R. Damaševičius, *Interpreting CNN for brain Tumor classification using XGrad-CAM*, In: Advanced Research in Technologies, Information, Innovation and Sustainability, ARTIIS 2024, Communications in Computer and Information Science, Springer, **2348** (2024). https://doi.org/10.1007/978-3-031-83435-6_21
13. S. Tehsin, I. M. Nasir, R. Damaševičius, *Explainability of disease classification from medical images using XGrad-CAM*, In: International conference on advanced research in technologies, information, innovation and sustainability, Springer, 2024, 221–236.
14. I. M. Nasir, S. Tehsin, R. Damaševičius, A. Zielonka, M. Woźniak, *Explainable cubic attention-based autoencoder for skin cancer classification*, In: Artificial Intelligence and Soft Computing, Lecture Notes in Networks and Systems, Springer, **15166** (2025). https://doi.org/10.1007/978-3-031-81596-6_11
15. S. Tehsin, I. M. Nasir, R. Damaševičius, Explainable brain tumor segmentation via attention-guided hybrid CNN–transformer–Mamba network, *Knowl.-Based Syst.*, **343** (2026), 116000. https://doi.org/10.1016/j.knosys.2026.116000
16. I. M. Nasir, H. Alshaya, S. Tehsin, W. Bouchelligua, Adaptive vision–language transformer for multimodal CNS tumor diagnosis, *Biomedicines*, **13** (2025), 2864. https://doi.org/10.3390/biomedicines13122864

17. A. Waqar, Intelligent decision support systems in construction engineering: An artificial intelligence and machine learning approaches, *Expert Syst. Appl.*, **249** (2024), 123503. https://doi.org/10.1016/j.eswa.2024.123503

18. A. Waqar, I. Othman, N. Shafiq, M. S. Mansoor, Applications of AI in oil and gas projects towards sustainable development: A systematic literature review, *Artif. Intell. Rev.*, **56** (2023), 12771–12798. https://doi.org/10.1007/s10462-023-10467-7

19. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.*, **9** (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

20. B. N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio, *N-BEATS: Neural basis expansion analysis for interpretable time series forecasting*, In: International Conference on Learning Representations (ICLR), 2020. Available from: https://openreview.net/forum?id=r1ecqn4YwB.

21. C. Challu, K. G. Olivares, B. N. Oreshkin, F. G. Ramirez, M. M. Canseco, A. Dubrawski, *NHiTS: Neural hierarchical interpolation for time series forecasting*, In: Proceedings of the AAAI Conference on Artificial Intelligence, **37** (2023), 6989–6997. https://doi.org/10.1609/aaai.v37i6.25854

22. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., *Attention is all you need*, In: Advances in Neural Information Processing Systems, **30** (2017). Available from: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

23. H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, et al., *Informer: Beyond efficient transformer for long sequence time-series forecasting*, In: Proceedings of the AAAI Conference on Artificial Intelligence, **35** (2021), 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325

24. H. Wu, J. Xu, J. Wang, M. Long, *Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting*, In: Advances in Neural Information Processing Systems, **34** (2021), 22419–22430. Available from: https://proceedings.neurips.cc/paper_files/paper/2021/file/bcc0d400288793e8bdcd7c19a8ac0c2b-Paper.pdf.

25. Y. Nie, N. H. Nguyen, P. Sinthong, J. Kalagnanam, *A time series is worth 64 words: Long-term forecasting with transformers*, In: International Conference on Learning Representations (ICLR), 2023. Available from: https://openreview.net/forum?id=Jbdc0vTOcol.

26. B. Lim, S. Ö. Arik, N. Loeff, T. Pfister, Temporal fusion transformers for interpretable multi-horizon time series forecasting, *Int. J. Forecasting*, **37** (2021), 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012

27. R. Caetano, J. M. Oliveira, P. Ramos, Transformer-based models for probabilistic time series forecasting with explanatory variables, *Mathematics*, **13** (2025), 814. https://doi.org/10.3390/math13050814

28. Y. Yamaguchi, I. Suemitsu, W. Wei, CITRAS: Covariate-informed transformer for time series forecasting, *IEEE Access*, **14** (2026). https://doi.org/10.1109/ACCESS.2026.3571646

29. L. Han, Y. Liu, Q. Deng, J. Jiang, Y. Sun, Z. Yu, et al., UniCA: Adapting time series foundation model to general covariate-aware forecasting, *arXiv Preprint*, 2025. Available from: https://arxiv.org/abs/2506.22039.

30. S. B. Punati, S. Kanta, U. B. Cheerala, M. G. Lanjewar, P. Damacharla, Temporal fusion transformer for multi-horizon probabilistic forecasting of weekly retail sales, *arXiv Preprint*, 2025. https://doi.org/10.48550/arXiv.2511.00552

31. S. Zhao, Y. Lin, X. Yang, Q. Lu, H. Xue, G. Jiang, Optimization of deep learning models for dynamic market behavior prediction, *arXiv Preprint*, 2025. https://doi.org/10.48550/arXiv.2511.19090

32. Z. Lyu, A. Ororbia, T. Desell, Online evolutionary neural architecture search for multivariate non-stationary time series forecasting, *Applied Soft Comput.*, **145** (2023), 110522. https://doi.org/10.1016/j.asoc.2023.110522

33. L. Su, X. Zuo, R. Li, X. Wang, H. Zhao, B. Huang, A systematic review for transformer-based long-term series forecasting, *Artif. Intell. Rev.*, **58** (2025), 80. https://doi.org/10.1007/s10462-024-11044-2

34. S. J. Taylor, B. Letham, Forecasting at scale, *Am. Stat.*, **72** (2018), 37–45. https://doi.org/10.1080/00031305.2017.1380080

35. D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, DeepAR: Probabilistic forecasting with autoregressive recurrent networks, *Int. J. Forecasting*, **36** (2020), 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001

AIMS Press