processes

MDPI

Article

Multi-Model Fusion Demand Forecasting Framework Based on Attention Mechanism

Chunrui Lei [1], Heng Zhang [1,*], Zhigang Wang [2] and Qiang Miao [1]

[1] College of Electrical Engineering, Sichuan University, Chengdu 610065, China; 2022223035096@stu.scu.edu.cn (C.L.); mqiang@scu.edu.cn (Q.M.)
[2] Jiangsu Sinoclouds S&T Co., Ltd., Zhenjiang 212000, China; lmfever@163.com
* Correspondence: hengzhang27@scu.edu.cn

Abstract: The accuracy of demand forecasting is critical for supply chain management and strategic business decisions. However, as data volumes grow and demand patterns become increasingly complex, traditional forecasting methods encounter significant challenges in processing intricate multi-dimensional data and achieving a satisfactory predictive accuracy. To address these challenges, this paper proposed an end-to-end multi-model demand forecasting framework based on attention mechanisms. The framework employs a dual attention mechanism to dynamically extract features from both the temporal and product dimensions, while integrating conditional information captured through convolutional neural networks, thereby enhancing its ability to model complex demand patterns. Additionally, a channel attention mechanism is introduced to perform the weighted fusion of outputs from multiple predictive models, thereby overcoming the limitations of single-model approaches and improving adaptability to varying demand patterns across diverse scenarios. The experimental results demonstrate that the proposed method outperforms conventional approaches across several evaluation metrics, achieving a 42% reduction in Mean Squared Error (MSE) compared to the baseline model. This notable improvement enhances both the accuracy and stability of demand forecasting. The framework offers valuable insights for addressing large-scale and complex demand patterns, providing guidance for precise decision-making and resource optimization within supply chain management. Future research will concentrate on further enhancing the model's generalization capability to manage missing data and demand fluctuations. Additionally, efforts will focus on integrating diverse heterogeneous data sources to assess its performance in various practical scenarios, ultimately improving the model's accuracy and flexibility.

Keywords: demand forecasting; dual attention; conditional information; multi-model fusion

Citation: Lei, C.; Zhang, H.; Wang, Z.; Miao, Q. Multi-Model Fusion Demand Forecasting Framework Based on Attention Mechanism. *Processes* **2024**, *12*, 2612. https://doi.org/10.3390/pr12112612

Academic Editor: Blaž Likozar

Received: 24 October 2024
Revised: 11 November 2024
Accepted: 19 November 2024
Published: 20 November 2024

1. Introduction

In today's increasingly competitive market environment, rapid shifts in consumer demand and heightened market volatility make it crucial for businesses to accurately track market dynamics and manage resources effectively. The demand for goods serves as a vital input for numerous business decisions, including upstream product procurement, local inventory management, and downstream logistics distribution activities [1]. As a bridge connecting market demand to internal business operations, demand forecasting influences not only short-term activities like inventory management and production scheduling but also plays a direct role in shaping long-term strategic planning and competitive positioning. Accurate demand forecasting enables companies to mitigate issues such as stockouts or overstock caused by demand fluctuations [2] while optimizing marketing strategies, including developing effective promotional campaigns and precisely tracking market trends. This ultimately enhances supply chain efficiency and improves customer satisfaction [3]. Therefore, improving the accuracy of demand forecasting is of critical importance for businesses.

Initially, demand forecasting research predominantly relied on historical sales data as input, utilizing univariate time series analysis methods for predicting future sales [4].

However, due to market uncertainties and the complexity of the data, product demand is influenced not only by historical sales figures but also by various external factors, such as seasonality, holiday promotions, and market trends [5]. For sales patterns affected by multiple factors, multivariate models tend to provide more accurate forecasting results. Moreover, the rise of e-commerce has led to a rapid increase in product variety, and market dynamics have become more volatile, with consumer purchasing behavior becoming more diversified [6]. As a result, accurately forecasting product demand has become a more challenging task.

Currently, the mainstream demand forecasting methods can be broadly categorized into statistical methods and computational intelligence approaches. Statistical methods mainly include Autoregressive Integrated Moving Average (ARIMA) models [7], the Prophet model [8], and Fourier analysis [9]. However, these linear models are limited in capturing unquantifiable factors, such as consumer psychology and market changes, and tend to perform poorly when dealing with complex nonlinear relationships. Furthermore, as the variety of products and data sources increases, companies face the challenge of processing vast amounts of real-time data, and traditional time series methods struggle with adaptability and scalability in the context of big data. As a result, computational intelligence methods, such as support vector machine (SVM) [10], artificial neural network (ANN) [11], and Long Short-Term Memory (LSTM) networks [12], have gained prominence in demand forecasting in recent years. These methods are proficient at learning complex nonlinear relationships and high-dimensional features and effectively managing long-term dependencies in sequential data. However, these models primarily focus on the temporal dimension and conditional information, frequently overlooking the interrelationships between different products. Additionally, a single forecasting model is insufficient for capturing the composite features of time series sales data and encounters notable limitations when dealing with complex features and long sequence dependencies. For instance, while LSTM and Gated Recurrent Unit (GRU) networks can capture long-term dependencies in sales time series, they tend to get trapped in local optima when handling multi-dimensional features and global information. To further improve forecasting performance, some researchers have explored composite models that integrate various time series analysis methods for more effective demand pattern modeling. Li et al. [13] enhanced the output of Gated Recurrent Unit (GRU) using an attention mechanism and integrated it with the Prophet model through objective-weighted fusion, effectively improving the accuracy of sales forecasting. Multi-model fusion enables the consolidation of strengths from different models, allowing for the capture of diverse patterns within demand data, particularly when demand is influenced by various factors such as seasonality and market fluctuations. Additionally, an effective fusion method can integrate multi-dimensional features, thereby enhancing the accuracy, robustness, and stability of demand forecasting. However, these current composite models typically combine time series analysis methods with machine learning approaches or integrate machine learning with deep learning methods, using multi-model fusion strategies that combine the outputs of individual models in a post-processing phase rather than through joint training within a unified framework. A key limitation of this approach is that it fails to fully leverage the synergies between models, particularly as each model emphasizes distinct aspects of data feature extraction. The simple post-fusion of results often overlooks the interactive information between models. Moreover, during model fusion, traditional methods such as objective weighting [13], genetic algorithm-based weighting [14], or weight allocation through other machine learning techniques all have their limitations. Objective weighting does not dynamically adapt to changes in data, whereas genetic algorithms and machine learning-based weighting methods, though capable of optimizing weights, necessitate secondary training, which leads to higher computational costs. Furthermore, the attention mechanism has been widely utilized in forecasting. In predictive tasks, it is predominantly employed to enhance feature selection [15] and manage global dependencies [16], with relatively few applications in multi-model fusion.

To address the poor adaptability of existing single-model demand forecasting methods in handling diverse demand signals, as well as the challenges related to weight selection, the lack of joint training capability, and high computational costs in multi-model fusion approaches, this paper proposes an end-to-end, attention-based multi-model fusion deep learning framework for demand forecasting. By leveraging the strengths of deep learning models, the proposed method effectively addresses the limitations of traditional approaches in joint training and weight allocation flexibility. The experimental results demonstrate that the proposed method outperforms traditional methods in complex scenarios, highlighting its promising application prospects. The specific contributions are as follows:

(1) A dual attention mechanism is designed to perform feature extraction from both the temporal and product dimensions, dynamically capturing correlations between different time points and products. This effectively improves the model's sensitivity to complex demand fluctuations. Conditional information (e.g., date, promotional factors, etc.) is extracted using convolutional neural networks and fused with the time and product features obtained from the dual attention mechanism to form a comprehensive feature extraction module. This ensures that the model simultaneously considers historical sales data and external conditions, thereby improving the comprehensiveness and accuracy of the predictions;
(2) This paper introduces a multi-model fusion strategy based on a channel attention mechanism, enabling joint training. By adaptively adjusting the contribution weights of each model, the proposed method overcomes the limitations of single models in multi-dimensional data forecasting, leveraging complementary strengths across the models and further improving the prediction performance;
(3) Experimental results demonstrate that the proposed framework outperforms conventional methods across multiple evaluation metrics, achieving a 42% reduction in MSE compared to the baseline LSTM forecasting model. A series of ablation experiments further validate the effectiveness of the framework, improving the accuracy of demand forecasting. This improvement facilitates more informed decision-making, offering valuable guidance for supply chain management activities.

The remainder of the paper is organized as follows: Section 2 reviews the relevant literature. Section 3 presents the proposed end-to-end multi-model fusion demand forecasting framework. Section 4 discusses the experimental results, including comparative experiments and ablation studies of the proposed framework. Finally, Section 5 summarizes the research findings and presents some conclusions.

2. Literature Review

Accurate demand forecasting is essential for optimizing supply chain management and strategic business decision-making [17]. As the data volume grows and demand patterns become increasingly complex, traditional statistical methods have demonstrated limitations in managing nonlinear and multi-dimensional data. In contrast, computational intelligence methods have become the mainstream approach in demand forecasting due to their strengths in feature extraction and recognizing complex patterns. Accordingly, this section briefly overviews both statistical and computational intelligence methods.

Time-related factors are often considered key variables influencing demand forecasting. Consequently, statistical methods, as traditional demand forecasting approaches, primarily depend on the time series analysis of historical data. Steinker et al. [18] integrated external factors such as promotions, holidays, and weather conditions (e.g., sunshine, temperature, and rainfall) into a demand forecasting model for retailers. They employed the Seasonal Autoregressive Integrated Moving Average with Exogenous Variables (SARIMAX) model and found that weather factors contributed to reducing forecast errors, significantly enhancing the accuracy of demand predictions. Nucamendi-Guillén et al. [19] employed an exponential smoothing model for demand forecasting in the fashion retail industry, assisting businesses in inventory and transportation management. Ramos et al. [20] conducted a case study of five different types of retail products, comparing the forecasting

performance of state-space models with that of the ARIMA model. Ma et al. [21] developed an Autoregressive Distributed Lag (ADL) model for forecasting using data from multiple product categories and stores, yielding favorable results. However, with rapid changes in market environments and the diversification of consumer behavior, demand fluctuations have become increasingly complex and nonlinear. The assumptions inherent in traditional statistical methods (such as linear relationships and stationarity) constrain their performance in managing multi-dimensional and nonlinear data. Consequently, although statistical methods can perform well in small-scale, short-term forecasts, they increasingly demonstrate limitations when addressing big data and complex demand patterns.

Computational intelligence methods, particularly those based on machine learning and deep learning, have demonstrated greater adaptability in demand forecasting. These methods can autonomously learn complex patterns, nonlinear relationships, and the interactions of high-dimensional features in data, making them especially effective for handling multi-dimensional time series and complex datasets. Craparotta et al. [22] employed a Siamese Neural Network (SNN) to enhance fashion sales forecasting by integrating heterogeneous data, including both image and time series data. Their case analysis yielded favorable results. Salinas et al. [23] proposed a probabilistic forecasting method using an Autoregressive Recurrent Neural Network model (DeepAR), which outperformed previous methods in forecasting accuracy. Abbasimehr et al. [24] developed a demand forecasting method based on a multi-layer LSTM network, enhancing prediction accuracy. Their experimental results indicated that this method surpassed other commonly used standard methods. Vallés-Pérez et al. [25] presented an architecture based on LSTM and transformer models, enabling the training model to adapt to different time steps and effectively addressing sales forecasting challenges.

Despite the successes of individual models in demand forecasting, they often show limitations when handling complex, multi-dimensional data. Single models often fail to capture the multi-level relationships and nonlinear features within the data, particularly when dealing with multi-dimensional condition information (e.g., promotions, temperature) and long time series data, which can negatively impact prediction performance. To overcome these limitations, researchers have increasingly adopted hybrid models that combine the strengths of multiple methods to further improve their prediction accuracy and robustness. Li et al. [13] developed a composite model with an attention mechanism that integrates the Gated Recurrent Unit (GRU) and Prophet models for sales forecasting. The model used the CRITIC method for weight assignment, making it suitable for rapidly changing market demands. Punia et al. [14] explored the relationships between various influencing factors and sales sequences, designing an ensemble method based on genetic algorithms to merge the strengths of deep learning and machine learning models. This approach outperformed individual models in short-term, mid-term, and long-term real-time demand forecasting. Ma et al. [26] introduced meta-learning for the first time in sales demand forecasting, proposing a meta-learner based on convolutional neural network (CNN) to assign weights to base predictors, demonstrating its effectiveness. Punia et al. [27] proposed a forecasting method based on a combination of random forests and Long Short-Term Memory (LSTM) networks, modeling complex temporal relationships and achieving a higher accuracy than other forecasting methods. However, these hybrid models are often designed by combining deep learning and machine learning models, and due to the heterogeneity between these two types of models, their training processes are usually not efficiently coordinated within a single framework. Furthermore, during the model fusion process, fixed weights or a weight optimization through genetic algorithms or other intelligent algorithms are often used. This not only increases computational time costs but also raises the risk of model overfitting.

In summary, while statistical methods excel at handling linear, well-structured data, they struggle with nonlinear and multi-dimensional complex data. On the other hand, computational intelligence methods, despite their significant advantages in feature extraction and complex pattern recognition, still face challenges in multi-model fusion and joint

training. To address these challenges, this paper proposes an end-to-end multi-model fusion deep learning demand forecasting framework based on an attention mechanism. This approach fully harnesses the strengths of deep learning models and successfully overcomes the limitations of traditional methods in joint training and flexible weight distribution.

3. Chapter Approaching the Problem

In this study on product demand forecasting, our goal is to predict the sales of a specific product over a defined future period, ranging from time $t+1$ to $t+predlength$, using data from the previous $windows$ time periods up to time t. In addition to considering the product's historical sales data, we also incorporate contextual factors such as temperature, promotions, and calendar time. Thus, the entire problem can be described by Equation (1).

$$\hat{y}_{t+1:t+predlength} = f(item_{1:j,(t-w):t}, condition_{1:k,(t-w):(t+predlength)}) \quad (1)$$

In this context, $\hat{y}_{t+1:t+predlength}$ represents the forecast target, which is the specific sales of the product from time $t+1$ to $t+predlength$. $f(\cdot)$ denotes the forecasting function, which in this study corresponds to the proposed demand forecasting framework. $item_{1:j,(t-w):t}$ represents the historical sales data of J products from time $t-w$ to time t, with w being the historical sliding window length. $condition_{1:k,(t-w):(t+predlength)}$ refers to the values of k types of contextual factors, such as temperature, promotions, and calendar time, from time $t-w$ to time $t+predlength$.

To enhance the effectiveness of the multi-model fusion and improve the accuracy of product demand forecasting, this paper proposes an end-to-end multi-model fusion framework based on product correlations, as illustrated in Figure 1. The framework is primarily composed of four components: data preprocessing, feature extraction and fusion, model learning, and model fusion and prediction.

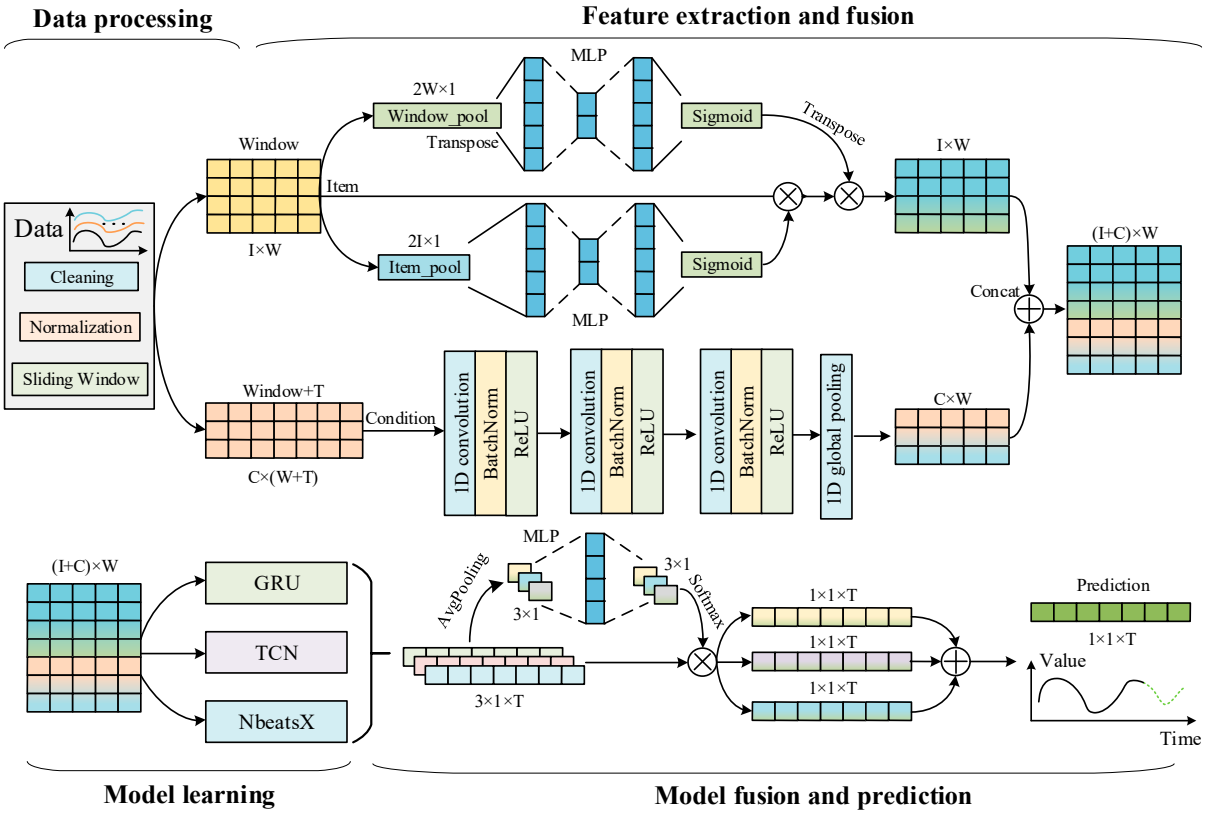

Figure 1. Multi-model fusion demand forecasting framework based on attention mechanism.

3.1. Data Preprocessing

In product-based demand forecasting, data preprocessing is a crucial foundational step that ensures both the effectiveness and the accuracy of the model. It consists mainly of three stages: data cleaning, data normalization, and sliding window processing. These preprocessing stages provide more accurate inputs for model learning, establishing a solid foundation for subsequent training and thereby guaranteeing the precision of demand forecasting.

(1) Data Cleaning

During the acquisition and transmission of product sales data, unforeseen issues may arise, resulting in anomalies such as redundancy and missing values. Therefore, data cleaning is essential to eliminate redundant and null values from the raw product sales data, thus enhancing the accuracy of the forecasts.

(2) Data Normalization

Product demand is often influenced by various factors such as seasonality, promotional activities, and market trends. Therefore, standardizing the data format and scale enables the model to better capture underlying relationships. By normalizing the sales data, discrepancies in sales volumes between different products can be eliminated, enabling the model to focus more on the relative demand fluctuations across products. Thus, data normalization is performed using Equation (2) to eliminate inconsistencies across different units of measurement.

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{2}$$

Here, x_i represents the ith data point of a certain feature and x_i' denotes the normalized data. x_{max} and x_{min} represent the maximum and minimum values, respectively, of all data points for that feature.

In addition, date-related features, such as the day of the year and the day of the month, are crucial conditional factors in demand forecasting. However, due to the discrete nature of dates, directly using these features may cause instability during model training. To address this issue, this study applies a sinusoidal transformation to the date information prior to normalization. This transformation converts the date features into continuous, periodic variables, smoothing the changes in date-related features and eliminating disruptions caused by abrupt shifts. This approach enables the model to better capture underlying temporal patterns and cyclical fluctuations, as shown in Equation (3).

$$x' = \frac{1}{T}\sin(2\pi x\frac{1}{T}) \tag{3}$$

where, x represents the date-related influencing factor, x' denotes the transformed sinusoidal date feature, and T refers to the period of the corresponding date feature.

(3) Sliding Window

The sliding window technique is a crucial step in demand forecasting, as it effectively captures trends and patterns in time series data. By creating windows of a fixed size, we can extract data features from consecutive time periods, enabling the model to better understand the influence of historical data on future demand. This approach not only enhances the local correlation within the data but also strengthens the model's predictive capability.

For the product information features, we utilize a fixed sliding window with a width of *window_size* to estimate and predict future sales. The prediction target corresponds to the product sales over the next *pred_length* time steps, with a sliding window stride of 1. The specific process is illustrated in Figure 2.

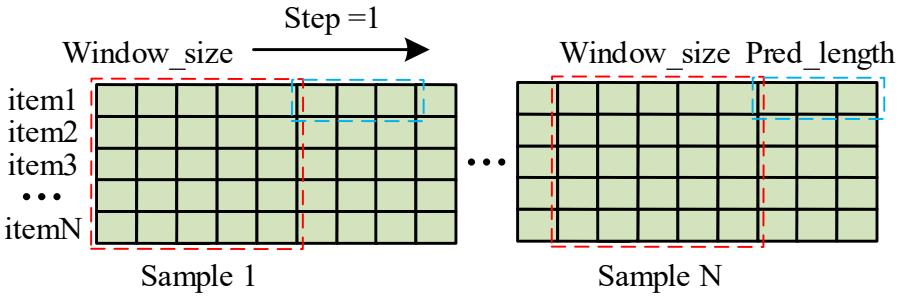

Figure 2. Sliding window processing of commodity features.

As for conditional information such as date features and promotional factors, since these factors can be anticipated in advance, we utilize a fixed sliding window with a width of *window_size* + *pred_length* for their estimation and prediction. The sliding window stride is set to 1, accordingly. The detailed process is shown in Figure 3.

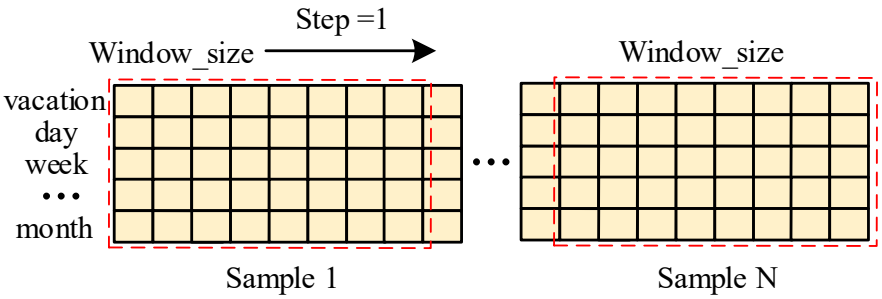

Figure 3. Sliding window processing of conditional information.

3.2. Feature Extraction and Fusion

3.2.1. Dual Attention Mechanism Based on Time and Product

Currently, most attention mechanisms, such as SE attention, primarily focus on calculating feature weights within a single dimension, often overlooking the complex interactions between multiple dimensions. In the field of product demand forecasting, most research predominantly emphasizes the time dimension of time series data, with limited in-depth analysis of the product dimension. Consequently, the interrelationships between different products are frequently overlooked, limiting the model's ability to capture demand fluctuations. To address these shortcomings, this paper introduces a dual attention mechanism that simultaneously considers both the time and product dimensions. By accounting for the significance of both dimensions, this mechanism offers a more comprehensive understanding of how various factors influence demand forecasting. It enables the exploration of inter-product relationships and provides more precise feature representations for model training. The attention structure takes the data tensor $X \in R^{I \times W}$ as input and produces an output tensor of the same size. The specific structure is illustrated in Figure 4.

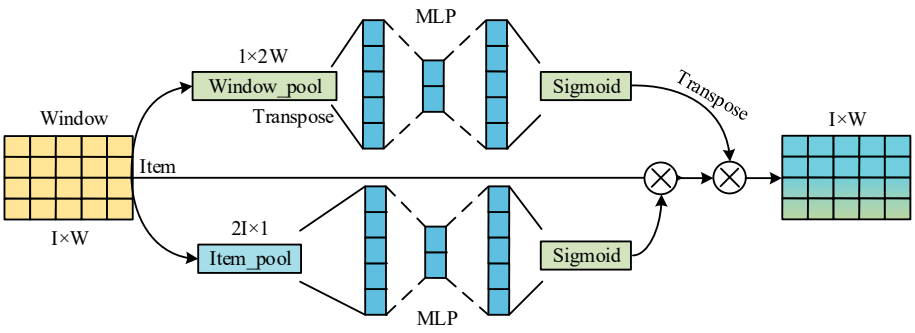

Figure 4. Dual attention based on time and item.

(1) Global information embedding

For the historical sales data stored as a two-dimensional tensor, we apply one-dimensional pooling operations along both the product and time dimensions. This compresses global information, while simultaneously encoding both product-specific and temporal information. To capture the global context and salient features of the feature maps, we utilize both global average pooling and max pooling, concatenating their outputs. This approach helps capture diverse feature representations across the product and time dimensions, while also reducing the risk of model overfitting.

The procedure is as follows: for an input $X \in R^{I \times W}$, vectors of sizes $(2I, 1)$ and $(1, 2W)$ are used to encode the historical sales data along the time and product dimensions, respectively. The aggregated output along the time dimension can then be expressed as

$$X_{W\_mean}(w) = \frac{1}{I}\sum_{i=0}^{I} x(i,w), X_{W\_max}(w) = \max(\sum_{i=0}^{I} x(i,w)) \tag{4}$$

$$X_{w\_pool} = concat(X_{w\_mean}, X_{w\_max}) \tag{5}$$

In the equation, $X_{W\_mean} \in R^{1 \times W}$ represents the vector obtained from global average pooling, $X_{W\_max} \in R^{1 \times W}$ represents the vector obtained from max pooling, and $X_{w\_pool} \in R^{1 \times 2W}$ is the aggregated output along the time dimension. Similarly, the aggregated output along the product dimension can be represented as follows:

$$X_{I\_mean}(i) = \frac{1}{W}\sum_{w=0}^{W} x(i,w), X_{I\_max}(i) = \max(\sum_{w=0}^{W} x(i,w)) \tag{6}$$

$$X_{I\_pool} = concat(X_{I\_mean}, X_{I\_max}) \tag{7}$$

In this context, $X_{I\_mean} \in R^{I \times 1}$ represents the vector obtained from global average pooling, $X_{I\_max} \in R^{I \times 1}$ denotes the vector derived from max pooling, and $X_{I\_pool} \in R^{2I \times 1}$ signifies the final aggregated output along the product dimension.

(2) Adaptive fusion correction

Building on the aforementioned compression operations, we propose a simple gating mechanism incorporating a fusion truncation function. This mechanism is designed to learn the nonlinear relationships between the aggregated vectors, making full use of the global and salient information in both the time and product dimensions. The gating structure consists of two fully connected layers: the first, a dimensionality reduction layer W_1, with a reduction ratio r, which helps reduce the computational burden, followed by a ReLU activation function and an upscaling fully connected layer W_2. Specifically, the aggregated vectors from Equations (5) and (7) are first transposed to match dimensions, then passed through the two fully connected layers for attention-based fusion and correction, as shown in Equations (8) and (9).

$$s_w = \sigma(W_2(\delta(W_1 X_{w\_pool}))) \tag{8}$$

$$s_I = \sigma(W_2'(\delta(W_1' X_{w\_pool}))) \tag{9}$$

In this framework, $s_w \in R^{1 \times W}$ represents the time-dimensional attention vector after transformation, $s_I \in R^{1 \times I}$ denotes the product-dimensional attention vector after transformation, δ is the nonlinear activation function ReLU, and σ is the Sigmoid activation function, which introduces the gating mechanism. After dimension matching, the final output of our dual attention mechanism, integrating both time and product dimensions, is as follows:

$$Y_1(i,w) = x(i,w) \times s_w(w) \times s_i(i) \tag{10}$$

Compared to other single-channel attention mechanisms, our dual attention module offers enhanced capabilities. It fully leverages both global and significant information from

the original time series data, while simultaneously computing attention weights across both the time and product dimensions. The result is a mask that matches the shape of the input data and contains the attention weights associated with each input feature, thereby providing robust support for subsequent demand forecasting.

3.2.2. Conditional Information Feature Extraction Based on Convolutional Neural Network

Convolutional neural network (CNN) can automatically extract and generate deep features from input time series data and images. With key principles such as local receptive fields, weight sharing, and pooling—distinct from traditional feedforward neural network—CNN demonstrate exceptional capability in feature extraction and robustness to variations in input data. As a result, they have shown superior performance in many machine learning and pattern recognition tasks.

Building upon this, for the conditional information in historical sales data (such as date features, promotional factors, etc.), we designed a three-layer convolutional neural network (CNN) architecture for feature extraction. This network is specifically aimed at fully leveraging the impact of these conditional factors on historical sales data, as illustrated in Figure 5.

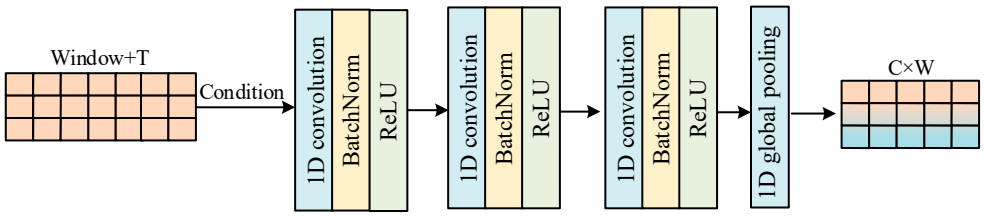

Figure 5. Conditional information feature extraction based on convolutional neural network.

The network architecture consists of three layers of 1D convolutional neural network (1D CNN), batch normalization layers, activation functions, and pooling operations. The input condition information is first passed through the 1D CNN layers for feature extraction, capturing local temporal patterns and variations in the condition data, such as periodicity in date features and abrupt changes due to promotional factors. To stabilize the distribution of network inputs and accelerate training, batch normalization layers are applied after each convolutional layer, helping to prevent gradient vanishing. The ReLU activation function is then used to introduce nonlinearity, enabling the model to capture complex features in the condition data. After the convolutional layers, a global average pooling layer is applied to reduce the number of parameters, retain critical information, prevent overfitting, and improve computational efficiency. Thus, after feature extraction and dimension matching through the three convolution layers, the condition input $X \in R^{C\times(W+T)}$ results in the feature output $Y_2 \in R^{C\times W}$. Here, C represents the condition dimension, W is the sliding window width, and T denotes the prediction length.

3.2.3. Feature Fusion

After feature extraction, we employed a feature fusion strategy to effectively integrate the time and product information extracted by the dual attention mechanism with the conditional feature information obtained through the convolutional neural network (Figure 6). This approach ensures that both the temporal and product-related dependencies, as well as the conditional factors, are comprehensively considered in the final feature representation for demand forecasting.

Specifically, through a concatenation operation, these two types of features are combined into a unified feature matrix, as shown in Equation (11). The significance of this feature fusion lies in the fact that the time and product information provide the model with a dynamic perspective on demand variations, while the conditional feature information, such as date features and promotional factors, adds contextual background. This

integration enhances the model's comprehensiveness and accuracy in subsequent demand forecasting tasks.

$$Y = concat(Y_1, Y_2) \tag{11}$$

where $Y \in R^{(I+C)\times W}$ represents the fused feature vector, $Y_1 \in R^{I\times W}$ denotes the time and product feature information extracted through the dual attention mechanism, and $Y_2 \in R^{C\times W}$ refers to the condition feature information extracted via the convolutional neural network. I represents the product dimension, C indicates the condition information dimension, and W is the sliding window width.

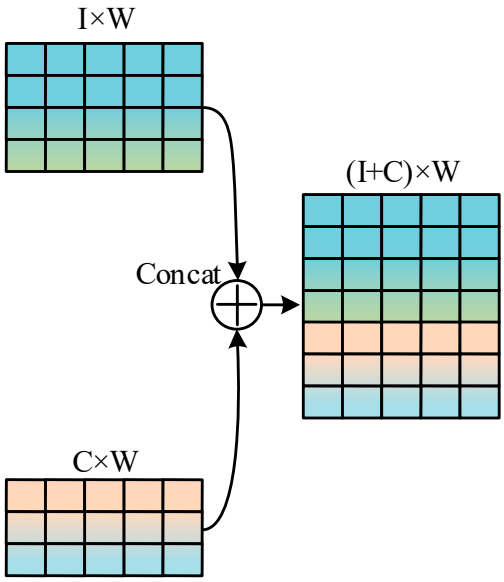

Figure 6. Feature fusion.

3.3. Model Learning

In time series forecasting, commonly used models include Recurrent Neural Network (RNN), Long Short-Term Memory network (LSTM), Gated Recurrent Unit (GRU), Temporal Convolutional Network (TCN), and the Nbeats model. Each of these models has unique characteristics, making them suitable for different types of time series data and demand patterns. RNN and their variants, LSTM and GRU, are particularly effective in capturing long-term dependencies in time series data. However, they often suffer from issues such as vanishing gradients when dealing with complex patterns, which can limit their ability to simultaneously capture both long-term and short-term trends. In contrast, TCN enhance the ability to extract local features through convolutional operations and address the gradient problems associated with traditional RNN. However, TCN may lack the flexibility of recurrent networks when modeling long-range dependencies. Nbeats, a relatively new framework, focuses on modeling trends and seasonality. While it performs well across various types of time series, it may not adapt as effectively to domain-specific requirements. In order to make this more intuitive and clear, we illustrate the advantages and disadvantages of commonly used prediction models in Table 1.

Due to the influence of various conditional factors and the rapid fluctuations in product sales, a single model often struggles to fully capture the intricate patterns within time series data. Consequently, predictions based solely on a single model may encounter limitations in both accuracy and stability. To address this limitation, we adopted a multi-model fusion forecasting strategy. As shown in Table 1, statistical models like ARIMA and machine learning models like random forest (RF) incur high computational costs due to their inability to be trained jointly. Recurrent Neural Network models, such as LSTM and GRU, capture long-term dependencies; however, the GRU has fewer parameters than LSTM, allowing for greater computational efficiency. The TCN model enables efficient parallel processing and enhances the capture of complex demand patterns through convolution

operations, though it is susceptible to overfitting. The Nbeats and NbeatsX models provide interpretability, focusing on trend and seasonality modeling; NbeatsX extends Nbeats by incorporating external variables, enhancing its versatility in handling diverse time series. While the Informer network delivers a high accuracy, it requires substantial data and computational resources.

Table 1. Model comparison.

Model	Advantages	Limitations
ARIMA	Easy to use, interpretable	Limited to univariate, cannot train jointly
RF	Suitable for nonlinear relationships	Struggles with temporal dependencies, no joint training
LSTM	Captures long-term dependencies, memory capability	High computational cost, risk of gradient vanishing
GRU	Captures long-term dependencies, efficient	Slightly lower accuracy
TCN	Captures long-term dependencies, efficient parallel processing	Lacks memory, prone to overfitting
Nbeats	Interpretable, leverages trends and seasonality	Limited to univariate
NbeatsX	Incorporates external variables over Nbeats	Prone to overfitting
Informer	High accuracy, strong adaptability	Requires large datasets and computational resources

Based on these observations, we implemented an ensemble approach that incorporated a GRU, TCN, and NbeatsX during model training. Leveraging the strengths of these three distinct model architectures, we achieved multi-level feature learning, enhancing the model's capacity to capture and interpret demand variations. The GRU effectively manages short- and mid-term dynamic changes, the TCN captures long-term dependencies, and NbeatsX offers robust trend and seasonality modeling. This ensemble approach not only improves the overall model performance but also enhances its adaptability to diverse data characteristics and demand patterns, providing robust support for accurate demand forecasting.

(1) Gated Recurrent Unit (GRU) [28]

The GRU is a widely adopted variant of the RNN designed to address the gradient vanishing problem commonly encountered in traditional RNNs when learning long sequences. It introduces a gating mechanism consisting of an update gate and a reset gate, which regulate the flow of information and control the forgetting process. Compared to LSTM, the GRU offers a simpler architecture and superior computational efficiency, making it particularly effective for handling time series data and tasks that require modeling long-term dependencies.

(2) Temporal Convolutional Network (TCN) [29]

The TCN is a sequence-modeling approach based on convolutional operations. It employs causal convolutions and dilated convolutions, which enable the model to capture long-term dependencies in sequential data. Unlike a traditional RNN, a TCN handles long-range dependencies by parallelizing computations and expanding the receptive field, without significantly increasing its computational complexity. This approach offers greater stability and enhanced training efficiency, making TCN particularly well suited for modeling time series data with long-term dependencies.

(3) NbeatsX [30]

NbeatsX is an extension of the Nbeats model, specifically developed for time series forecasting. It employs a fully feedforward neural network-based stacked module structure,

which progressively extracts multi-level features from time series data. By incorporating residual connections, the model enhances its prediction accuracy. NbeatsX is highly flexible, capable of adapting to various trends and seasonal patterns in time series data, and it has garnered significant attention for its exceptional predictive accuracy and interpretability.

3.4. Model Fusion and Prediction

To achieve model fusion and prediction, conventional methods often combine the outputs of multiple models using simple weighted averages or optimization algorithms to assign weights, facilitating information sharing and complementarity among models. However, these approaches have several limitations. First, the weights in conventional methods are typically fixed, and their selection is challenging, preventing the model from fully accounting for performance variations across different tasks. This leads to the underestimation of certain model outputs during fusion, particularly when handling complex demand fluctuations where key features may not be effectively captured. Second, these methods lack dynamic adjustment mechanisms, rendering them unable to adapt to changes in input data or fluctuations in model performance. In practical applications, where data characteristics and environmental factors often change, fixed weight allocation is ineffective, diminishing the accuracy and reliability of predictions.

To overcome these limitations, we introduce the Squeeze-and-Excitation (SE) attention mechanism [31] to adaptively assign weights to different models, improving the model fusion process, as shown in Figure 7. Compared to self-attention and other attention mechanisms, the Squeeze-and-Excitation (SE) attention mechanism involves a simple two-step operation, without the need for complex similarity calculations or attention matrix generation. This makes it more computationally efficient in handling multi-model outputs, maintaining performance while reducing the computational burden. Moreover, SE offers improved adaptability in multi-model fusion. By employing the SE attention mechanism, we can dynamically compute the weights of each model's output, reflecting its relative importance in the current forecasting task. This approach not only fully leverages feature information from each model but also adaptively adjusts the weights through joint training, allowing the system to better handle complex demand patterns and dynamic fluctuations.

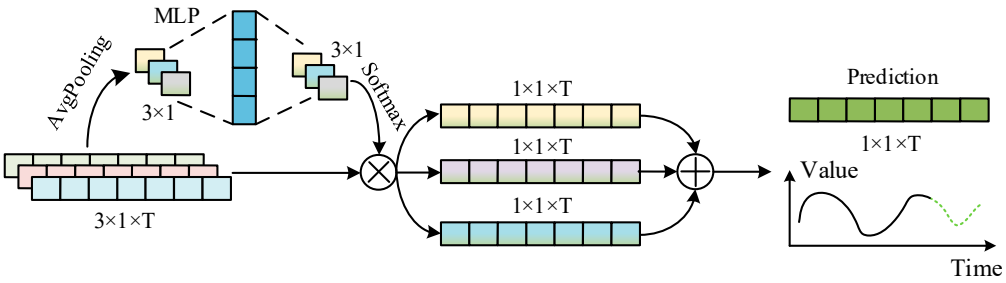

Figure 7. Model fusion and prediction.

Specifically, the output of each model undergoes global average pooling to extract global feature information. A fully connected layer is then applied to transform these features and generate corresponding weight coefficients, reflecting the relative importance of each model in the current forecasting task. These computed weights are applied to the outputs of the individual models, and the weighted outputs are summed to obtain the final prediction. This approach effectively integrates feature information from multiple models, enhancing the accuracy and robustness of the predictions. By dynamically adjusting the model weights, we not only capitalize on the strengths of each individual model but also mitigate errors that may arise from relying on a single model, ultimately leading to more precise and reliable demand forecasting.

4. Experimental Analysis

In this section, we present the dataset and experimental parameters used in our study. We then conduct experiments to compare the proposed model framework with commonly used time series forecasting methods, demonstrating the superiority of our approach. Finally, ablation experiments are performed to assess the effectiveness and necessity of each module. All tests were conducted on a server equipped with three NVIDIA GeForce RTX 4090 GPUs, running CentOS Stream 9 as the operating system.

4.1. Data Description

The dataset used in this study is sourced from an electronics-manufacturing company and spans sales data from 20 stores over the period of 2013 to 2018. It includes sales records for five similar products over six years, resulting in a high-dimensional time series dataset with an extensive temporal range. The dataset not only contains the daily sales volume of each product across different stores but also records various external factors, such as temperature, holidays, and promotional events, that influence sales. This rich set of contextual information allows for a more comprehensive analysis of how external factors impact product demand.

4.2. Evaluation Index

To quantitatively assess the performance of the proposed method, we employed several evaluation metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R^2), and Mean Absolute Percentage Error (MAPE), for a comprehensive evaluation of the forecasting model's accuracy and robustness. These metrics were employed to evaluate the model's performance in demand forecasting.

First, MSE is used to evaluate the model's performance by calculating the squared differences between predicted and actual values, as shown in Equation (12). MSE is particularly sensitive to larger errors, making it effective at capturing significant deviations during demand peaks, particularly in cases of dramatic fluctuations, such as promotional seasons or holidays. A lower MSE indicates that the model effectively handles sudden, large demand variations and provides more stable and accurate forecasts, which supports inventory control and resource optimization during periods of demand volatility.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{12}$$

Here, y_i represents the actual values, $\hat{y}_i$ denotes the predicted values, and n is the number of samples.

Next, MAE is employed to evaluate the model by calculating the absolute difference between predicted and actual values, providing a direct measure of forecast bias, as shown in Equation (13). MAE provides an intuitive error metric, reflecting the overall accuracy of the model. It focuses on the model's stability in handling routine demand variations and measures the long-term precision of demand forecasting.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{13}$$

R^2 is employed to assess the model's ability to explain the variance in the data. The value of R^2 ranges from 0 to 1, with values closer to 1 indicating the better fit of the model. As shown in Equation (14), R^2 reflects the model's adaptability to demand patterns, including long-term trends and cyclical fluctuations, providing insight into how well the model captures these demand dynamics.

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{14}$$

In the equation, $\overline{y}$ represents the mean of the actual values.

Finally, MAPE is employed to evaluate the model's accuracy by calculating the percentage of the forecast error relative to the actual values, as shown in Equation (15). MAPE reflects the consistency of the model's accuracy across different products and time periods, particularly when dealing with products subject to significant demand fluctuations. A lower MAPE indicates that the model maintains a high accuracy across varying demand levels.

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{15}$$

A lower MSE and MAE indicate a reduction in overall forecast error, enabling better demand prediction, minimizing inventory overstocking, and reducing stockouts. This ultimately reduces inventory costs and enhances customer satisfaction. A lower MAPE reflects the relative accuracy of predictions, assisting supply chain decision-makers in forecasting demand more accurately for different times and products. This helps prevent over-purchasing or stock shortages, thereby improving inventory turnover and minimizing waste.

4.3. Comparative Experiment

To assess the effectiveness of the proposed method, a series of comparative experiments were conducted, comparing the model presented in this study with widely adopted and state-of-the-art models in demand forecasting. Specifically, seven benchmark models were selected: ARIMA, LSTM, GRU, TCN, 1DCNN, Nbeats, and Informer. Python 3.7.16 was used for the implementation, with the ARIMA model optimized using the Statsmodels package 0.14.0. The deep learning models were developed using the PyTorch 1.12.1 framework. The Adam optimizer [32] was chosen for the network, as it combines the advantages of momentum and adaptive learning rates, effectively addressing the sparse gradient problem and accelerating convergence. Adam has demonstrated its superiority in various deep learning applications, particularly in training large-scale datasets and complex models. The batch size was set to 64, the learning rate was set to 0.001, and the loss function used was the MSE loss function. The sliding window width was set to 30 days to predict the sales for the next 7 days. The specific experimental settings for comparison are shown in Table 2.

Table 2. Comparison method parameter.

Method	Parameter Settings
ARIMA	Set using the Auto-ARIMA method.
LSTM	num_layers: 2 hidden_size: 32
GRU	num_layers: 2 hidden_size: 32
TCN	kernel_size = 2 dropout = 0.2
1DCNN	kernel_size = 3 dropout = 0.3
NbeatsX	hidden_size = 128 num_stacks = 4 num_block = 3 stack_types = trend
Informer	n_head = 8 d_layers = 2
Proposed	window attention ratio = 3 product attention ratio = 2 SE_ratio = 2 conv_kernel_size = 3 conv_dropout = 0.3

To ensure fairness, each experiment was repeated five times, and the average of the evaluation metrics was computed as the final result, as presented in Table 3. Among them, the best performing model is shown in bold.

Table 3. Comparative experiment.

Methods/Metrics	MSE	MAE	R^2	MAPE
ARIMA	25.0069 ± 0.7798	3.6044 ± 0.0542	0.6900 ± 0.0097	7.94% ± 0.14%
LSTM	13.8058 ± 0.8859	2.7600 ± 0.1448	0.8879 ± 0.0073	6.93% ± 0.46%
GRU	13.7575 ± 1.1017	2.6828 ± 0.2161	0.8884 ± 0.0097	6.62% ± 0.61%
TCN	13.1931 ± 2.1062	2.5333 ± 0.2567	0.8924 ± 0.0179	6.05% ± 0.59%
1DCNN	22.8854 ± 13.3470	3.4789 ± 0.8482	0.8109 ± 0.1157	8.71% ± 2.01%
NbeatsX	11.3605 ± 0.5053	2.2191 ± 0.4140	0.9080 ± 0.1071	5.31% ± 0.11%
Informer	10.9094 ± 1.3002	2.3805 ± 0.1098	0.9031 ± 0.0202	6.73% ± 0.91%
Proposed	**7.9426 ± 0.7011**	**1.9665 ± 0.2007**	**0.9326 ± 0209**	**4.95% ± 0.52%**

As shown in Table 1, the traditional time series analysis method, ARIMA, struggles to effectively capture the complex dynamic relationships in such intricate, nonlinear sales data. Additionally, it is highly sensitive to outliers, resulting in a poorer performance, with a significantly lower predictive accuracy compared to the other models. Meanwhile, the one-dimensional convolutional neural network primarily relies on local feature extraction to capture patterns in the data, which makes it challenging to effectively learn the temporal dependencies and long-term trends, leading to a decrease in predictive accuracy. In contrast, the three widely used deep learning models—LSTM, GRU, and TCN—are better equipped to handle long-term dependencies within time series data and address the complex, nonlinear relationships in demand fluctuations. As a result, these models exhibit a relatively good and similar performance in the prediction results. The NbeatsX model, based on Nbeats, introduces conditional information features and utilizes a feedforward neural network with stacked fully connected layers, effectively capturing trends and seasonality in the input sequences, yielding satisfactory results. The Informer network, built on the transformer architecture, overcomes the limitations of traditional models in handling long sequences by leveraging sparse attention mechanisms and efficient hierarchical structures. In this case, it demonstrates a strong predictive performance. The network model proposed in this study incorporates a dual attention mechanism that comprehensively extracts temporal information, product characteristics, and external conditions, followed by an attention mechanism for multi-model fusion, which improves the overall prediction performance and accuracy. Therefore, it outperforms all the other models in this study. To present the prediction results more clearly and avoid clutter from excessive lines, only the top three performing models are selected for visualization, as shown in Figure 8.

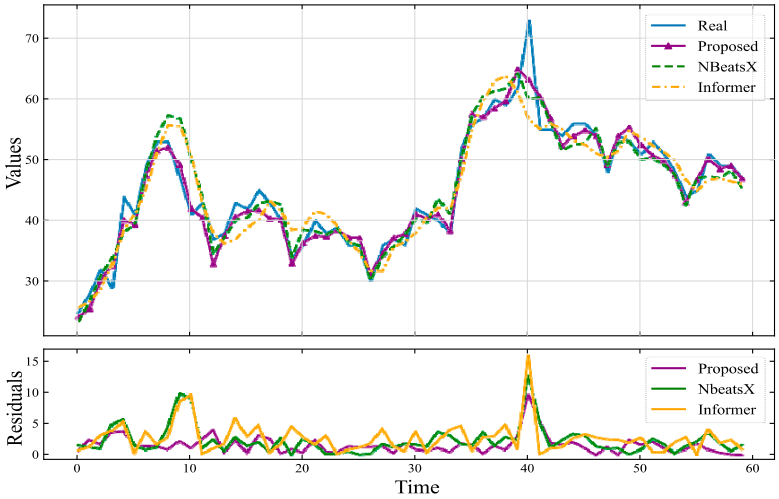

Figure 8. Comparison of the models.

The figure clearly indicates that, among the three models, the model proposed in this study aligns more closely with the actual sales values. Furthermore, the residual plot presented below offers a more intuitive perspective, demonstrating that our model exhibits relatively smaller errors than the other models.

4.4. Ablation Experiment

To validate the effectiveness of each module within the proposed model, we conducted a series of ablation experiments for each component and conducted comparative analyses. Furthermore, to assess the effect of different feature sets on the prediction accuracy, we performed feature ablation experiments.

4.4.1. Comparison of Feature Extraction Effectiveness

To illustrate the effectiveness of the proposed feature extraction structure for product demand forecasting, we designed a series of experiments to comprehensively assess the performance of this module. We designated the automated feature extraction module introduced in this study as "AF" and applied it to the benchmark models for comparative analysis. The specific experimental results are summarized in Table 4. Among them, the best performing model is shown in bold.

Table 4. Comparison of feature extraction effectiveness.

Methods/Metrics	MSE	MAE	R^2	MAPE
LSTM	13.8058	2.7600	0.8879	6.93%
GRU	13.7575	2.6828	0.8884	6.62%
TCN	13.1931	2.5333	0.8924	6.05%
NbeatsX	11.3605	2.2191	0.9080	5.31%
AF-LSTM	11.3468	2.1725	0.9096	5.33%
AF-GRU	10.8248	2.0595	0.9122	5.11%
AF-TCN	11.1251	2.2047	0.9100	5.25%
AF-NbeatsX	9.8425	1.9954	0.9213	5.03%
Proposed	**7.9426**	**1.9265**	**0.9326**	**4.95%**

The experimental results indicate that incorporating the feature extraction module designed in this study yields varying degrees of improvement in the predictive performance of the benchmark models, thereby validating its effectiveness. This feature extraction module first employs a dual attention mechanism to adaptively capture significant information, emphasizing the influence of various time points and products on the predictions. Subsequently, it employs a convolutional neural network to process conditional information and facilitate feature fusion, thereby enhancing the model's performance and effectively capturing the complex relationships between time, product, and conditional factors.

4.4.2. Comparison of Effectiveness of Multi-Model Fusion

To examine the advantages of multi-model fusion compared to single models, we designed a series of experiments comparing three individual benchmark models with our proposed attention-based multi-model fusion method (excluding the feature extraction module), as shown in Table 5. Among them, the best performing model is shown in bold.

Table 5. Comparison of effectiveness of fusion modules.

Methods/Metrics	MSE	MAE	R^2	MAPE
GRU	13.7575	2.6828	0.8884	6.62%
TCN	13.1931	2.5333	0.8924	6.05%
NbeatsX	11.3605	2.2191	0.9080	5.31%
Fusion module	**9.9736**	**2.1246**	**0.9175**	**5.02%**

The experimental results demonstrate that the model employing the attention mechanism for fusion surpasses the individual benchmark models across all performance metrics. By assigning dynamic weights to the outputs of various models, the fusion model effectively captures the complex relationships between time and product, thereby improving its overall predictive accuracy.

4.4.3. Comparison of Multi-Model Fusion Methods

To assess the effectiveness of the attention-based multi-model fusion, we conducted comparative experiments using four distinct methods: simple averaging, weighted averaging, fully connected network fusion, and attention mechanism fusion (Figure 9). In the simple averaging approach, equal weights were assigned to the three models. However, as shown in Table 1, the NbeatsX network exhibits superior performance compared to both the GRU and TCN. Consequently, in the weighted averaging approach, higher weights were assigned to NbeatsX, distributed as 4, 3, and 3 for NbeatsX, the GRU, and the TCN, respectively. Additionally, to further explore the advantages of attention weighting in processing multi-model outputs, we included a comparison group that utilized only fully connected fusion. Specifically, we initially vertically concatenated the outputs of the three models, followed by employing a fully connected layer to perform linear combinations and achieve dimension matching for the final prediction results.

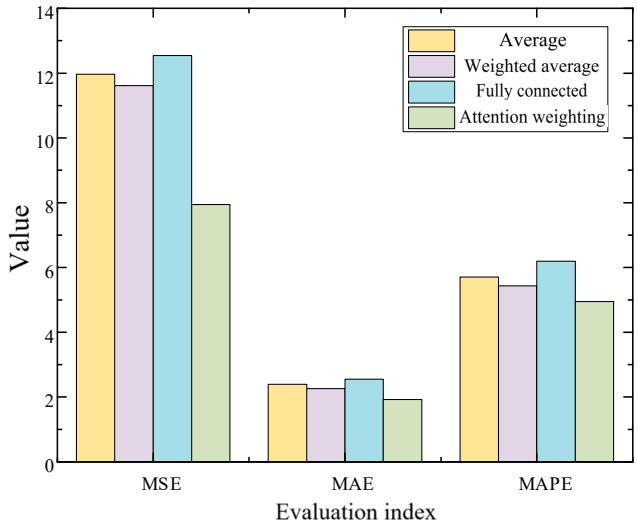

Figure 9. Comparison of multi-model fusion methods.

The experimental results demonstrate that among various fusion strategies, attention-weighted fusion achieves the highest performance. In comparison to simple and weighted averaging, the integration of the attention mechanism substantially improves the prediction accuracy. This finding confirms the effectiveness of attention mechanisms in processing multi-model outputs, facilitating a more precise evaluation of each model's significance. Notably, although weighted averaging performs better than simple averaging, its fusion results remain inferior to those of the top-performing model, NbeatsX. This implies that an improper weight distribution can negatively impact prediction accuracy, further emphasizing the benefits of an adaptive weight allocation via attention mechanisms. Furthermore, the fully connected fusion demonstrated the lowest performance, suggesting that simple concatenation and linear combinations did not effectively utilize the information from multi-model outputs.

4.4.4. Feature Sensitivity Analysis

To evaluate the impact of different feature sets on model accuracy, this section categorizes the feature set into three components: the predicted product, additional product information, and the effect of conditional information, including date, temperature, and

promotion. These features will be sequentially removed to assess their impact on the model's performance (Table 6). Among them, the best performing model is shown in bold.

Table 6. Feature sensitivity analysis.

Feature/Metrics	MSE	MAE	R2	MAPE
Target product	10.3399	2.2049	0.8819	4.89%
Other products	8.4191	2.1069	0.9072	4.55%
Date	8.7960	2.2030	0.8995	5.13%
Temperature	8.3863	2.1365	0.9057	4.45%
Promotion	10.6666	2.2141	0.8788	4.99%
Include all features	**7.9426**	**1.9265**	**0.9326**	**4.95%**

The experimental results show that different feature sets impact the model accuracy to varying degrees. Removing features related to the target product and promotional information had the most pronounced effect on accuracy. Excluding the target product's historical sales data weakens the predictive capability, as the sales trend and seasonality of the product are essential predictors. Promotional information is also a crucial factor, as sales fluctuations often correlate with promotional activities. The removal of promotional information reduces accuracy, particularly during promotional periods. Excluding information about other products limits the model's ability to capture inter-product relationships, reducing its ability to predict ripple effects in sales. Excluding date information diminishes the model's awareness of seasonality and time trends, which may reduce accuracy for cyclical patterns. Omitting temperature information has a comparatively minor impact, indicating that the product's sales are relatively insensitive to temperature changes. Notably, the removal of individual features does not significantly degrade model performance, highlighting the robustness of the proposed method.

The experimental results show that the proposed model generally performs well across scenarios but may face challenges in highly complex demand patterns. Significant shifts in data characteristics or external factors may impair the model's ability to capture such changes effectively, thereby reducing its predictive accuracy. In such cases, incorporating additional features or employing more complex models could improve performance, though it may increase model complexity and computational costs. Although our method demonstrates effectiveness within the scope of this study, its adaptability and scalability across various sectors and domains require further validation. Distinct demand patterns, industry characteristics, or external factors may require model retraining or adaptation to maintain efficacy and robustness in different contexts. Furthermore, because demand patterns vary over time, periodic model updates or retraining may be necessary, particularly when the data distribution changes significantly. Future research could focus on enhancing model adaptability to reduce the need for frequent retraining.

5. Conclusions

In the context of today's rapidly evolving market environment, accurate demand forecasting has emerged as a crucial tool for businesses aiming to optimize inventory management, enhance supply chain efficiency, and formulate precise marketing strategies. Effective demand forecasting not only reduces inventory costs but also enhances customer satisfaction and responsiveness to market changes. However, traditional forecasting models demonstrate considerable limitations in feature extraction and information integration, leading to inadequate performance in capturing complex market dynamics and consumer behavior. These challenges critically undermine the accuracy and reliability of demand predictions, thereby necessitating innovative solutions from researchers. To tackle this challenge, this study proposes a feature extraction framework that integrates a dual attention mechanism with a Squeeze-and-Excitation (SE) attention mechanism for multi-model fusion. The design of the dual attention mechanism enables the dynamic weighting of features from both the temporal and product dimensions, emphasizing the information most

critical for demand forecasting. This approach not only enhances the model's adaptability but also facilitates the effective capture of the interactions between various time points and products. Furthermore, the SE attention mechanism allows for the flexible integration of outputs from different models based on their significance. During the implementation process, we initially employ the dual attention mechanism to extract key features, followed by the use of convolutional neural networks to process relevant conditional information. Ultimately, the extracted features are effectively integrated through a multi-model fusion architecture to generate the final prediction results. The experimental results indicate that the proposed method outperforms traditional approaches across multiple evaluation metrics, thereby validating the effectiveness of both the dual attention mechanism and the multi-model fusion strategy.

By effectively integrating the strengths of various models, the end-to-end deep learning demand forecasting framework proposed in this study provides valuable insights for future forecasting in complex scenarios. Furthermore, it assists businesses in optimizing inventory management and supply chain operations, promoting agile responses and informed decision-making in dynamic environments, thereby enhancing their competitive advantage. Nevertheless, certain limitations exist. First, the accuracy of deep learning models heavily depends on historical data, potentially limiting their prediction accuracy when sample sizes are small or data are missing. Additionally, this study primarily considers time series and conditional information as feature inputs, without incorporating more heterogeneous data sources, such as social media feedback and macroeconomic indicators, which may also influence demand fluctuations. This limitation may reduce the model's applicability in specific contexts, indicating the need for further scalability improvements. Future research should aim to enhance the model's generalization capability, enabling it to handle missing data and fluctuating demand scenarios. Additionally, incorporating diverse data sources and multiple demand-influencing factors will be explored to assess the model's performance in various real-world contexts, ultimately enhancing both its accuracy and its flexibility.

Author Contributions: Conceptualization, C.L.; methodology, C.L.; software, C.L.; validation, C.L.; formal analysis, C.L.; investigation, Q.M. and Z.W.; resources, Z.W.; data curation, C.L.; writing—original draft preparation, C.L.; writing—review and editing, H.Z.; visualization, C.L.; supervision, Q.M. All authors have read and agreed to the published version of the manuscript.

Funding: The research was partially supported by the National Key R&D Program of China (No. 2021YFB3300800, No. 2021YFB3300801 and No. 2021YFB3300803).

Data Availability Statement: Due to privacy or ethical restrictions, the data supporting the reported results cannot be publicly shared.

Conflicts of Interest: Author Zhigang Wang was employed by the Jiangsu Sinoclouds S&T Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Punia, S.; Singh, S.P.; Madaan, J.K. A cross-temporal hierarchical framework and deep learning for supply chain forecasting. *Comput. Ind. Eng.* **2020**, *149*, 106796. [CrossRef]
2. Güven, I.; Şimşir, F. Demand forecasting with color parameter in retail apparel industry using artificial neural networks (ANN) and support vector machines (SVM) methods. *Comput. Ind. Eng.* **2020**, *147*, 106678. [CrossRef]
3. Wolters, J.; Huchzermeier, A. Joint in-season and out-of-season promotion demand forecasting in a retail environment. *J. Retail.* **2021**, *97*, 726–745. [CrossRef]
4. Ulrich, M.; Jahnke, H.; Langrock, R.; Pesch, R.; Senge, R. Distributional regression for demand forecasting in e-grocery. *Eur. J. Oper. Res.* **2021**, *294*, 831–842. [CrossRef]
5. Babai, M.Z.; Boylan, J.E.; Rostami-Tabar, B. Demand forecasting in supply chains: A review of aggregation and hierarchical approaches. *Int. J. Prod. Res.* **2022**, *60*, 324–348. [CrossRef]
6. Kharfan, M.; Chan, V.W.K.; Firdolas Efendigil, T. A data-driven forecasting approach for newly launched seasonal products by leveraging machine-learning approaches. *Ann. Oper. Res.* **2021**, *303*, 159–174. [CrossRef]

7. Tarmanini, C.; Sarma, N.; Gezegin, C.; Ozgonenel, O. Short term load forecasting based on ARIMA and ANN approaches. *Energy Rep.* **2023**, *9*, 550–557. [CrossRef]
8. Borges, D.; Nascimento, M.C. COVID-19 ICU demand forecasting: A two-stage Prophet-LSTM approach. *Appl. Soft Comput.* **2022**, *125*, 109181. [CrossRef]
9. Yukseltan, E.; Yucekaya, A.; Bilge, A.H. Hourly electricity demand forecasting using Fourier analysis with feedback. *Energy Strategy Rev.* **2020**, *31*, 100524. [CrossRef]
10. Jiang, P.; Li, R.; Liu, N.; Gao, Y. A novel composite electricity demand forecasting framework by data processing and optimized support vector machine. *Appl. Energy* **2020**, *260*, 114243. [CrossRef]
11. Seyedan, M.; Mafakheri, F. Predictive big data analytics for supply chain demand forecasting: Methods, applications, and research opportunities. *J. Big Data* **2020**, *7*, 53. [CrossRef]
12. Joseph, R.V.; Mohanty, A.; Tyagi, S.; Mishra, S.; Satapathy, S.K.; Mohanty, S.N. A hybrid deep learning framework with CNN and Bi-directional LSTM for store item demand forecasting. *Comput. Electr. Eng.* **2022**, *103*, 108358. [CrossRef]
13. Li, Y.; Yang, Y.; Zhu, K.; Zhang, J. Clothing sale forecasting by a composite GRU–Prophet model with an attention mechanism. *IEEE Trans. Ind. Inform.* **2021**, *17*, 8335–8344. [CrossRef]
14. Punia, S.; Shankar, S. Predictive analytics for demand forecasting: A deep learning-based decision support system. *Knowl.-Based Syst.* **2022**, *258*, 109956. [CrossRef]
15. Tian, C.; Niu, T.; Wei, W. Developing a wind power forecasting system based on deep learning with attention mechanism. *Energy* **2022**, *257*, 124750. [CrossRef]
16. Hu, Y.; Xiao, F. Network self attention for forecasting time series. *Appl. Soft Comput.* **2022**, *124*, 109092. [CrossRef]
17. Miklós-Thal, J.; Tucker, C. Collusion by algorithm: Does better demand prediction facilitate coordination between sellers? *Manag. Sci.* **2019**, *65*, 1552–1561. [CrossRef]
18. Steinker, S.; Hoberg, K.; Thonemann, U.W. The value of weather information for e-commerce operations. *Prod. Oper. Manag.* **2017**, *26*, 1854–1874. [CrossRef]
19. Nucamendi-Guillén, S.; Moreno, M.A.; Mendoza, A. A methodology for increasing revenue in fashion retail industry: A case study of a Mexican company. *Int. J. Retail Distrib. Manag.* **2018**, *46*, 726–743. [CrossRef]
20. Ramos, P.; Santos, N.; Rebelo, R. Performance of state space and ARIMA models for consumer retail sales forecasting. *Robot. Comput.-Integr. Manuf.* **2015**, *34*, 151–163. [CrossRef]
21. Ma, S.; Fildes, R.; Huang, T. Demand forecasting with high dimensional data: The case of SKU retail sales forecasting with intra-and inter-category promotional information. *Eur. J. Oper. Res.* **2016**, *249*, 245–257. [CrossRef]
22. Craparotta, G.; Thomassey, S.; Biolatti, A. A siamese neural network application for sales forecasting of new fashion products using heterogeneous data. *Int. J. Comput. Intell. Syst.* **2019**, *12*, 1537–1546. [CrossRef]
23. Salinas, D.; Flunkert, V.; Gasthaus, J.; Januschowski, T. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecast.* **2020**, *36*, 1181–1191. [CrossRef]
24. Abbasimehr, H.; Shabani, M.; Yousefi, M. An optimized model using LSTM network for demand forecasting. *Comput. Ind. Eng.* **2020**, *143*, 106435. [CrossRef]
25. Vallés-Pérez, I.; Soria-Olivas, E.; Martínez-Sober, M.; Serrano-López, A.J.; Gómez-Sanchís, J.; Mateo, F. Approaching sales forecasting using recurrent neural networks and transformers. *Expert Syst. Appl.* **2022**, *201*, 116993. [CrossRef]
26. Ma, S.; Fildes, R. Retail sales forecasting with meta-learning. *Eur. J. Oper. Res.* **2021**, *288*, 111–128. [CrossRef]
27. Punia, S.; Nikolopoulos, K.; Singh, S.P.; Madaan, J.K.; Litsiou, K. Deep learning with long short-term memory networks and random forests for demand forecasting in multi-channel retail. *Int. J. Prod. Res.* **2020**, *58*, 4964–4979. [CrossRef]
28. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
29. Lea, C.; Flynn, M.D.; Vidal, R.; Reiter, A.; Hager, G.D. Temporal convolutional networks for action segmentation and detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 156–165.
30. Olivares, K.G.; Challu, C.; Marcjasz, G.; Weron, R.; Dubrawski, A. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *Int. J. Forecast.* **2023**, *39*, 884–900. [CrossRef]
31. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
32. Kingma, D.P. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.