Soft Computing (2024) 28:12281–12297
https://doi.org/10.1007/s00500-024-09937-z

APPLICATION OF SOFT COMPUTING

Deep learning-driven intelligent pricing model in retail: from sales forecasting to dynamic price optimization

Dongxin Li[1] · Jiayue Xin[2]

Accepted: 1 May 2024 / Published online: 26 July 2024

Abstract

Under the wave of the digital era, the retail industry is facing unprecedented fierce competition and a rapidly changing market environment. In this context, developing smart and efficient pricing strategies has become a top priority in the industry. Faced with this challenge, traditional pricing methods are inadequate due to their slow response, insufficient adaptability to instant changes in the market, and over-reliance on historical data and human experience. In response to this urgent need, this study aims to design an intelligent pricing model rooted in deep learning to enhance the vitality and competitiveness of the retail industry. The emerging solution adopted in this article combines Temporal Fusion Transformer (TFT), Ensemble of Simplified RNNs (ES-RNN), and dynamic attention mechanisms, aiming to accurately capture and analyze complex time series data through these advanced technologies. TFT processes multivariate and multi-level data, ES-RNN technology integrates multiple simple versions of recurrent neural networks to enhance predictive power, and the dynamic attention mechanism allows the model to dynamically weight the importance of different points in the time series, thereby improving the effectiveness of feature extraction. Test experimental results on four different data sets show that our models all show excellent performance, and the accuracy of predicted product sales far exceeds traditional models. In addition, with its ability to dynamically adjust pricing, the model demonstrates excellent stability and adaptability amid market fluctuations. This research not only promotes the intelligent transformation of retail pricing strategies, but also provides a more strategic tool for enterprises to compete for market share.

Keywords Deep learning · Retail industry · Smart pricing · Dynamic pricing · Predictive modeling · Data analysis

1 Introduction

In today's digital era, the retail industry faces increasingly fierce competition and changing market conditions. To stand out in a highly competitive market, retailers need to continuously innovate and adopt advanced technologies to improve business efficiency (Sujith et al. 2022). Among them, pricing strategy, as one of the core elements of the retail industry, directly affects sales, profits and market share. In order to better adapt to the complexity and dynamics of the market, deep learning technology has gradually become an important tool to improve the intelligence of pricing strategies. With the continuous accumulation of data in the retail industry and the rapid development of deep learning technology, we focus on developing an intelligent pricing model based on deep learning, and are committed to comprehensive research from sales forecasting to dynamic price optimization (Sunny et al. 2020). Traditional pricing methods often rely on experience and historical data, and are difficult to cope with the real-time fluctuations of the market and the changing characteristics of commodities. Therefore, by introducing deep learning technology, we hope to build a more intelligent and flexible pricing model to more accurately predict product sales and flexibly adjust pricing strategies while monitoring market changes in real time.

Currently, research on pricing strategies in the retail industry mainly focuses on traditional statistical models and rule-based methods (Du and Li 2019). In order to have a deeper understanding of the application of deep learning

✉ Jiayue Xin
32021120069@cueb.edu.cn

Dongxin Li
dorislee202209@163.com

1 School of Northeast Asia Studies, Shandong University, Weihai 264209, Shandong, China

2 School of Statistics, Capital University of Economics and Business, Beijing 100026, China

in the field of pricing, we will focus on four classic deep learning models: convolutional neural network (CNN), long short-term memory network (LSTM), gated recurrent unit (GRU) and Transformer Model. These models have achieved remarkable results in other fields. Next, their application and advantages in the field of intelligent pricing in the retail industry will be introduced.

CNN is a deep learning model specially designed to process grid data such as images and videos. The basic concept is to extract local features of the input data through convolutional layers and pooling layers. In this way, CNN can effectively capture the spatial relationships and hierarchical structures in the image (Liang et al. 2022). Among image processing tasks, CNN is known for its excellent performance in areas such as image classification, object detection, and segmentation. CNN has a wide range of application scenarios, from image recognition to autonomous driving, all of which demonstrate its excellent ability to process complex visual data. In the retail industry, the application of CNN can play a key role in product image recognition, packaging design analysis, etc. (Bao et al. 2021). By training deep convolutional networks, we can automatically extract product appearance, logos and features to provide retailers with more accurate product information. However, CNN also faces some challenges in smart pricing in the retail industry. First of all, retail data usually contains multiple dimensions, including image information and time series data, etc., and CNN is relatively difficult to model multi-dimensional data (Gao et al. 2023). Secondly, CNN is relatively weak in processing time series data, and it is difficult to capture long-term dependencies in time series.

LSTM is a variant of RNN that aims to solve the problem of traditional RNN's difficulty in capturing long-term dependencies on long sequence data (Mujeeb et al. 2019). Its core principle includes three gate control structures: forget gate, input gate and output gate. These gated structures allow LSTM to selectively retain or forget information when processing sequence data, effectively handling long-distance dependencies in time series. LSTM has achieved remarkable achievements in many fields, especially in natural language processing, speech recognition and time series analysis (Falatouri et al. 2022). Its superior performance on sequence data makes it one of the preferred models for time series modeling tasks. In smart pricing in the retail industry, LSTM can be applied to the modeling and prediction of sales data. By learning from historical sales data, LSTM is able to capture the impact of sales trends, seasonal changes and promotional activities on sales (Khalid et al. 2020). Although LSTM performs well in processing long sequence data, it still has some problems. First, LSTM may face vanishing or exploding gradient problems, especially when dealing with very long sequences. Secondly, the computational cost of LSTM is relatively high and is not suitable for scenarios with high real-time requirements.

GRU is an improved structure of RNN, designed to solve the vanishing gradient problem in LSTM, and is more computationally efficient (Li et al. 2021). The core principles of GRU include update gates and reset gates. The function of these two gates is to control the flow of information and process long-distance dependencies in sequences. GRU has achieved good results in many fields, especially in tasks such as speech recognition, machine translation, and image generation (Ruan et al. 2022). Compared with LSTM, GRU has fewer parameters, is more computationally efficient, and is easier to train. In smart pricing in the retail industry, GRU is widely used in time series modeling of sales data, which can effectively capture dynamic changes and trends in product sales. Although GRU has certain advantages in processing time series data, it still faces some challenges. Due to its relatively simple structure, GRU may not perform as well as LSTM in handling long-term dependencies. In smart pricing in the retail industry, we need to consider the long-term relationships that may exist in product sales. Therefore, this article will comprehensively consider GRU and other models to build a more comprehensive and flexible smart pricing model in the retail industry (Arora and Balyan 2023).

The transformer model is a sequence modeling architecture based on the self-attention mechanism, proposed by Vaswani et al. in 2017. The core principle is to use the self-attention mechanism to model the dependencies of different positions in the sequence, so that the model can better capture the global information in the sequence. Transformer has achieved remarkable results in the field of natural language processing, especially in machine translation tasks, outperforming traditional recurrent neural network structures (Xiao et al. 2022). In addition to natural language processing, Transformer also shows strong versatility in image processing, speech recognition and other fields (Gupta et al. 2022). In smart pricing in the retail industry, Transformer's global dependencies enable it to better capture the complex relationships between items, especially when processing sales data with multi-dimensional information. However, transformer also has some problems, one of which is its high computational cost, especially for longer sequences (Soleimani and Kezunovic 2020). In addition, in some time series data, Transformer may pay too much attention to past information and ignore future trends. It may be possible to improve detection accuracy through multi-strategy improvements, such as sparrow search algorithm and Gaussian pyramid, and deep transfer learning network (Chen et al. 2023). The method is also expected to enable multi-level classification to guide model optimization (Shao et al. 2022).

Although CNN, LSTM, GRU, and Transformer models have achieved great success in the field of deep learning, they have shown some shortcomings in smart pricing in the retail industry. First, they cannot fully take into account the multidimensional characteristics of sales data and cannot effectively capture the complex relationships between commodities. Secondly, these models may not perform satisfactorily in some cases when it comes to handling long-term and short-term time series relationships. In addition, there are certain challenges in terms of computing efficiency and real-time performance. Based on the shortcomings of the above work, this paper proposes the ensemble temporal net model which integrates multiple components to process time series data in smart pricing in the retail industry in a more comprehensive and flexible way. The ensemble temporal net model is mainly composed of the fusion of the temporal fusion transformer and the ensemble of simplified RNNs model framework, and introduces a dynamic attention mechanism for model optimization. Each component has its unique role: TFT (Temporal Fusion Transformer) is responsible for processing multi-variable, multi—for hierarchical time series data, ES-RNN (Ensemble of Simplified RNNs) improves the generalization performance of the model by integrating multiple simplified RNN models, while the dynamic attention mechanism allocates different time points or items based on the dynamics of sales data. Attention weight improves the model's sensitivity to key information. The advantage of the ensemble temporal net model is that it comprehensively utilizes the characteristics of each component and overcomes the limitations of the traditional model. By more comprehensively considering the diversity of sales data, this model can more accurately predict the sales of items at different price levels, thereby providing retailers with smarter and more flexible pricing strategies. This innovative model is of great significance for improving sales efficiency and enhancing market competitiveness.

The main contributions of this study are as follows:

- Deep learning model integration: The ensemble temporal net model cleverly integrates TFT, ES-RNN and dynamic attention mechanism. It helps to more comprehensively capture features in multi-dimensional sales data and improve the model's modeling ability of different commodities, prices and time changes.
- Multi-level time series modeling: The ensemble temporal net model achieves effective modeling of long-term and short-term time series relationships in sales data through the ES-RNN component. Helps more accurately capture sales trends, seasonal changes, and the impact of promotional activities on sales, providing more accurate information for intelligent pricing decisions.
- Introduction of a dynamic attention mechanism: A dynamic attention mechanism is introduced to allocate attention weights to different time points or products according to the dynamics of sales data. It can focus on important information more flexibly and improve sensitivity to key elements in sales data. properties, enhancing the adaptability and practicality of the model.

In the following chapters, we will expand the discussion according to the following structure: Section 2 will introduce the methods in depth and reveal the core construction and design principles of the ensemble temporal net model. Section 3 will focus on the experimental settings and details (Experiment) in order to reproduce the experiment. Section 4 will introduce the experimental results (Results) in detail and show the performance of the ensemble temporal net model in different data sets and scenarios. Finally, Sect. 5 will summarize and conclude the full text.

2 Methods

2.1 Overview of our network

The ensemble temporal net model is a deep learning framework proposed in this article, aiming to solve the sales forecast problem in intelligent pricing in the retail industry. The model is mainly composed of three key components, namely TFT, ES-RNN and dynamic attention mechanism. These components collaborate with each other to process multi-variable, multi-level time series data in a comprehensive and flexible manner, improving the model's prediction accuracy and generalization performance. TFT is mainly responsible for processing multi-variable and multi-level time series data. Through multi-level feature extraction and fusion, TFT can capture different levels of patterns in sales data and provide more comprehensive information for the model. ES-RNN improves the model's modeling capabilities for time series data by integrating multiple simplified RNN models. Its integrated nature helps improve stability and generalization performance, allowing the model to better adapt to different commodities and market environments. The dynamic attention mechanism is mainly responsible for allocating dynamic attention weights to different time points or products based on the dynamics of sales data. This mechanism allows the model to focus on important information more flexibly and increase sensitivity to key elements in sales data.

The network construction process of ensemble temporal net model is divided into the following steps: First, the input layer design takes multi-dimensional information of sales data as input, including product characteristics, price levels, timestamps, etc. The data is preprocessed and then input into the model; then the TFT component performs multi-level temporal feature extraction and fusion on input data to obtain different levels of pattern information. This component uses a self-attention mechanism to process multi-variable time series; then the ES-RNN component integrates multiple simplified RNN models to improve the generalization performance of the model by modeling the time series relationship of sales data. ES-RNN aggregates the prediction results of multiple RNNs through a voting mechanism. In addition, the dynamic attention mechanism component assigns dynamic attention weights to different time points or products. This component uses the learned weights to make the model more flexibly focus on important information in the sales data. Finally, there is the output layer design, which outputs the information processed by multiple components into the final sales forecast result through a suitable summary method.

The overall model structure diagram is shown in Fig. 1.

The running process of the Ensemble TemporalNet model is shown in Algorithm 1.

Algorithm 1 Ensemble TemporalNet Training

```
Input: Training dataset 𝒟
Output: Trained Ensemble TemporalNet model parameters θ
Initialize model parameters θ randomly
Split 𝒟 into training and validation sets
Initialize empty ensemble predictions list 𝒫
for each component model do
    Train component model on the training set
    Obtain component model predictions on the validation set: 𝒫_i
    Add 𝒫_i to 𝒫
end for
Train dynamic attention mechanism using 𝒫 and ground truth labels
Initialize TFT and ES-RNN model parameters
while not converged do
    Sample a batch from the training set
    Calculate temporal features using TFT
    Extract temporal representations using ES-RNN
    Calculate dynamic attention weights using the trained attention mechanism
    Combine temporal representations with attention weights
    Make final sales prediction using the combined information
    Calculate RMSE and MAE loss
    Update parameters using backpropagation
end while
```

The proposal of the ensemble temporal net model is of great significance to the topic of this article. First, through multi-component integration, the model can more comprehensively consider the diversity of sales data and improve its adaptability to different commodities and market environments. Secondly, the dynamic attention mechanism enables the model to focus on important information more flexibly and improve sensitivity to key elements in sales data, thereby improving prediction accuracy. Overall, the ensemble temporal net model provides a comprehensive and flexible solution for intelligent pricing in the retail industry, which is expected to improve sales efficiency and enhance market competitiveness in practical applications.

2.2 Temporal fusion transformer model

TFT is a deep learning model based on the self-attention mechanism, specially designed to process multi-variable and multi-level time series data (Theodoridis and Tsadiras 2022). Its basic principles include using the self-attention mechanism to capture key information in the sequence, and effectively capturing patterns at different levels in the sequence through multi-level feature extraction and fusion (Metin et al. 2023). The main purpose of TFT is to model and predict time series data, and is especially suitable for application scenarios of multi-variable time series such as sales data and meteorological data.

TFT has significant application advantages in sales forecasting in the retail industry. Since retail data usually contains multiple variables (such as product characteristics, prices, promotions, etc.) and multi-level time series relationships, TFT can capture this information more comprehensively and improve the sensitivity of the model to sales trends. Its self-attention mechanism allows the model to adaptively focus on the relationship between different time points and variables, making predictions more accurate.

In the ensemble temporal net model, TFT undertakes the task of multi-level feature extraction and fusion of sales data. By integrating TFT with other components, the entire model can better understand complex patterns in sales data and improve its adaptability to different commodities and market environments. The ability of TFT enables the ensemble temporal net model model to more comprehensively and flexibly predict sales in intelligent pricing in the retail industry, making an important contribution to improving the overall performance of the model.

The structure diagram of the TFT model is shown in Fig. 2.

The main formula of TFT is as follows:

$$X_t = \left[x_{t-L+1}, x_{t-L+2}, \ldots, x_{t-1}, x_t\right] \tag{1}$$

where X_t represents the input sequence at time t, $x_{t-L+1}, x_{t-L+2}, \ldots, x_{t-1}, x_t$ represents the historical observations from time $(t - L + 1)$ to t.

$$M_t = \sigma\left(W_m * X_t\right) \tag{2}$$

where M_t represents the memory vector at time t, $*$ represents the convolution operation, σ represents the activation function (e.g., Sigmoid), W_m represents the learnable convolutional filters.

$$C_t = \sigma\left(W_c * X_t\right) \tag{3}$$

where C_t represents the Constant vector at time t, $*$ represents the Convolution operation, σ represents the Activation

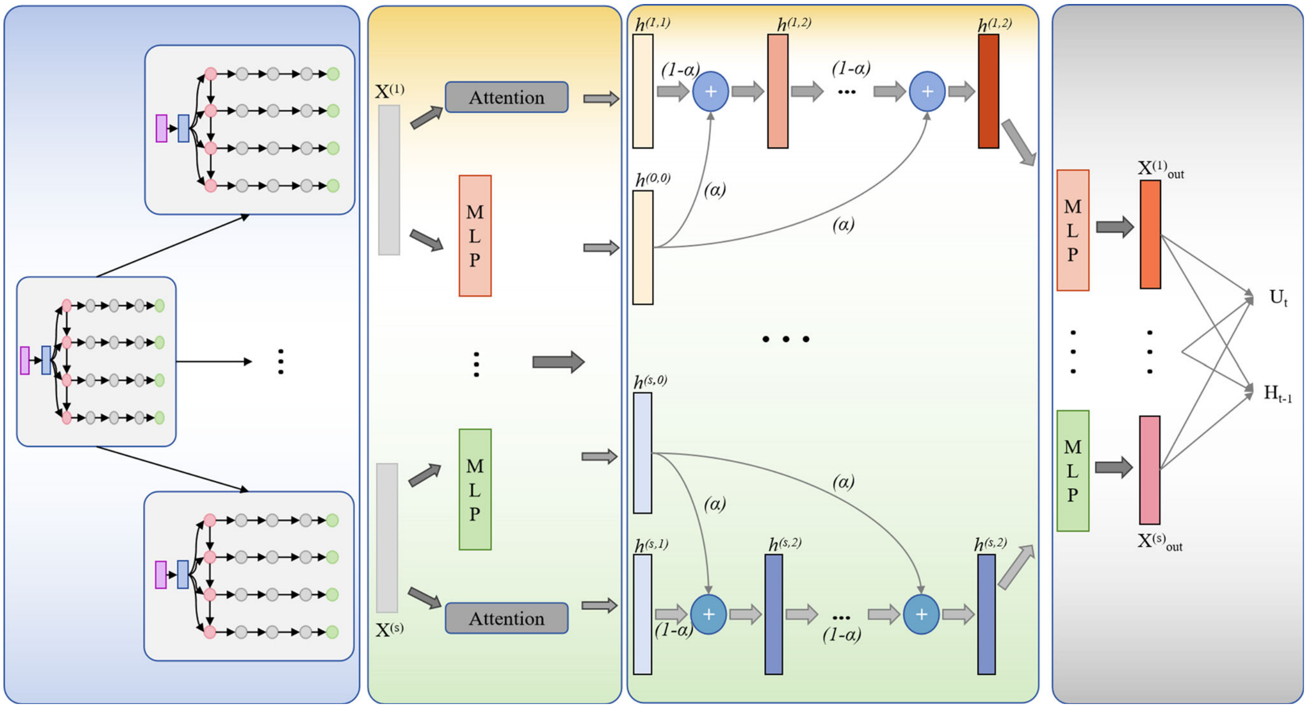

Fig. 1 Overall flow chart of the model

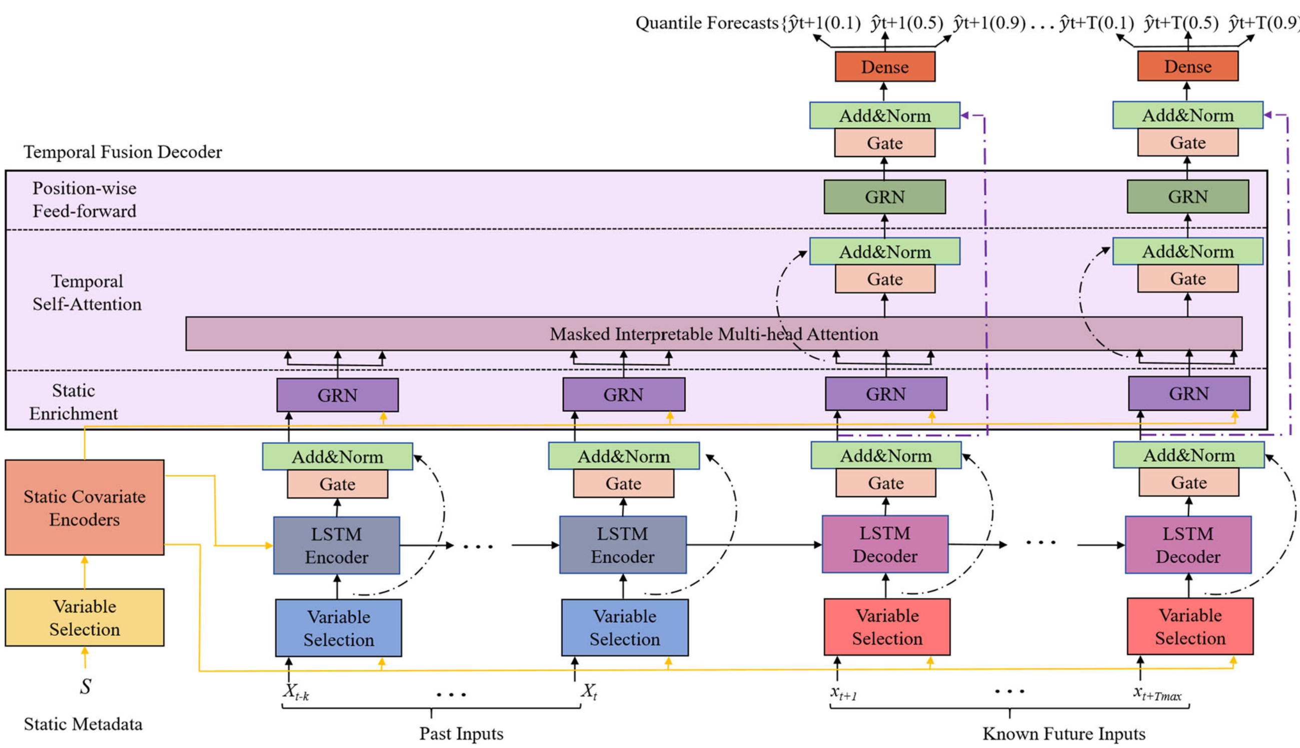

Fig. 2 Flow chart of the TFT model

function (e.g., Sigmoid), W_c represents the Learnable convolutional filters.

$$G_t = \sigma\left(W_g * X_t\right) \tag{4}$$

where G_t represents the gate vector at time t, $*$ represents the convolution operation, σ represents the activation function (e.g., Sigmoid), W_g represents the learnable convolutional filters.

$$S_t = \mathrm{softmax}\left(W_s * X_t\right) \tag{5}$$

where S_t represents the seasonal pattern vector at time t, $*$ represents the convolution operation, softmax represents the softmax activation function, W_s represents the learnable convolutional filters.

$$H_t = M_t \odot G_t + (1 - G_t) \odot (C_t \odot S_t) \tag{6}$$

where H_t represents the hidden state vector at time t, $\odot$ represents the element-wise multiplication.

$$O_t = \mathrm{Linear}(H_t) \tag{7}$$

where O_t represents the output vector at time t, Linear represents the linear transformation.

2.3 Ensemble of simplified RNNs model

ES-RNN is an ensemble model based on RNN (Recurrent Neural Network), designed to improve the modeling capabilities of time series data. The basic principle includes integrating multiple simplified RNN models to enhance the stability and generalization performance of the model. The main use of ES-RNN is to model and forecast time series data (Godahewa et al. 2021). In this article, it is particularly suitable for forecasting sales in the retail industry.

ES-RNN has shown significant advantages in applications in the field of smart pricing in the retail industry. Since retail data often has complex temporal relationships, ES-RNN integrates multiple simplified RNN models to enable the model to better capture potential patterns in sales data (Semenoglou et al. 2023). Its integrated nature helps improve the generalization performance of the model, allowing the model to achieve more stable prediction results under different commodities and market environments.

In the ensemble temporal net model, ES-RNN is responsible for modeling the time series relationship of sales data. By integrating multiple simplified RNN models, ES-RNN improves the model's modeling ability for sales data and enhances the generalization of the model. Its role in the overall model is to provide the model with more comprehensive and stable time series modeling capabilities, making the ensemble temporal net model more competitive in smart pricing in the retail industry.

The structure diagram of the ES-RNN model is shown in Fig. 3.

The main formula of ES-RNN is as follows:

$$H_t = \mathrm{ReLU}\left(W_h \cdot X_t + U_h \cdot H_{t-1} + b_h\right) \tag{8}$$

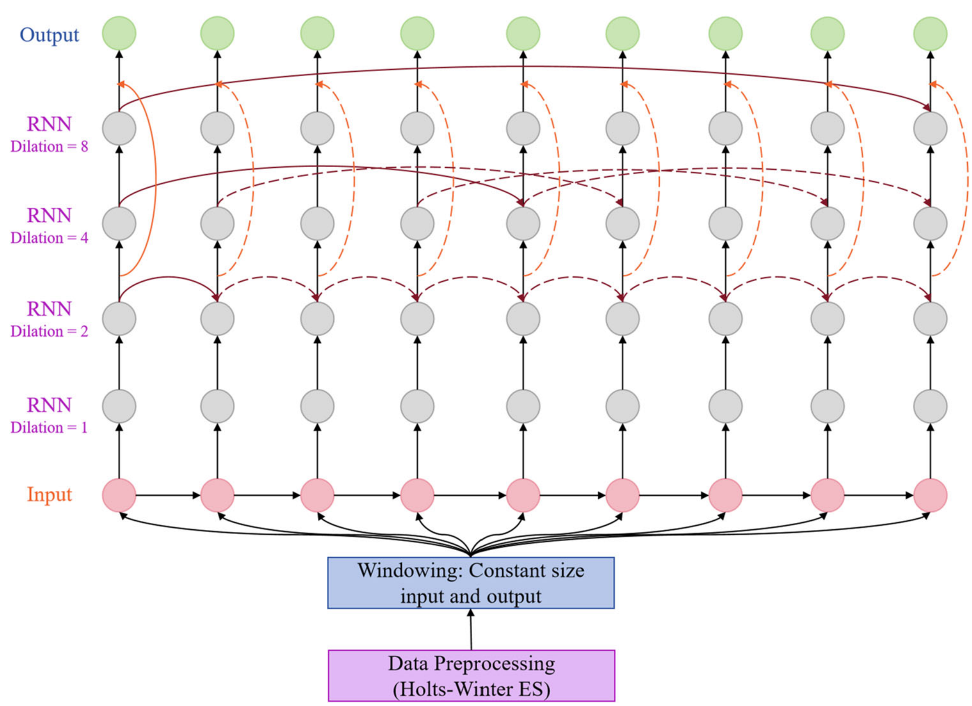

Fig. 3 Flow chart of the ES-RNN model

where H_t is the hidden state vector at time t, X_t is the input sequence at time t, W_h is the learnable weights for the input, U_h is the learnable weights for the hidden state, b_h is the bias term, ReLU is the rectified Linear Unit activation function.

$$O_t = \text{Linear}(H_t) \tag{9}$$

where O_t is the output vector at time t, H_t is the hidden state vector at time t, Linear is the linear transformation.

$$L_t = \text{Linear}(H_t) \tag{10}$$

where L_t is the output for the level component at time t, H_t is the hidden state vector at time t, Linear is the linear transformation.

$$S_t = \text{Softplus}(L_t) \tag{11}$$

where S_t: Smoothed level component at time t, L_t: Output for the level component at time t, Softplus: Softplus activation function.

$$E_t = \text{Linear}(H_t) \tag{12}$$

where E_t is the output for the error component at time t, H_t is the hidden state vector at time t, Linear is the linear transformation.

$$\sigma_t = \text{Sigmoid}(E_t) \tag{13}$$

where σ_t is the sigmoid activation for the error component at time t, E_t is the output for the error component at time t, Sigmoid is the sigmoid activation function.

$$Y_t = S_t \cdot O_t + (1 - S_t) \cdot Y_{t-1} \tag{14}$$

where Y_t is the predicted output at time t, S_t is the smoothed level component at time t, O_t is the output vector at time t, Y_{t-1} is the predicted output at time $(t-1)$.

2.4 Dynamic attention mechanism

Dynamic-AM is a model based on the attention mechanism, which aims to assign dynamic attention weights to different time points or variables based on the dynamics of sequence data (Xiao et al. 2021). The basic principle includes adaptively adjusting the attention weight according to the content of the input sequence data, so that the model can focus on important information more flexibly (Wang et al. 2021). In this article, the main purpose of Dynamic Attention Mechanism is to provide dynamic attention adjustment for the EnsembleTemporalNet model to better adapt to the temporal characteristics of sales data.

Dynamic-AM plays an important role in the field of intelligent pricing in the retail industry. Since sales data is dynamic, including promotions, seasonal changes, etc., Dynamic-AM can adaptively adjust attention weights, allowing the model to more flexibly capture the importance of different time points or products. Its advantage is that it improves the model's sensitivity to the dynamic characteristics of sales data, thereby enhancing the model's predictive ability.

In the ensemble temporal net model, the role of Dynamic Attention Mechanism is to provide the model with dynamic attention weights to adapt to temporal changes in sales data. By introducing dynamic attention adjustment into the overall model, this module allows ensemble temporal net to focus on important information more flexibly, improving the model's modeling capabilities for sales data. Its function in the overall model is to make the model more adaptable to the timing characteristics of the retail industry, providing key support for the optimization of intelligent pricing models.

The structure diagram of the Dynamic-AM is shown in Fig. 4.

The main formula Dynamic-AM is as follows:

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{T} \exp(e_{t,j})} \tag{15}$$

where $\alpha_{t,i}$ is the attention weight for the i-th element at time t, $e_{t,i}$ is the attention score for the i-th element at time t, T is the number of elements in the sequence.

$$e_{t,i} = v_a^T \tanh\left(W_a \cdot h_t + U_a \cdot x_i + b_a\right) \tag{16}$$

where $e_{t,i}$ is the attention score for the i-th element at time t, v_a is the learnable attention vector, W_a is the learnable weight matrix for the hidden state, U_a is the learnable weight matrix for the input element, h_t is the hidden state at time t, x_i is the representation of the i-th element in the sequence, b_a is the bias term.

$$c_t = \sum_{i=1}^{T} \alpha_{t,i} \cdot x_i \tag{17}$$

where c_t is the context vector at time t, $\alpha_{t,i}$ is the attention weight for the i-th element at time t, x_i is the representation of the i-th element in the sequence.

$$\tilde{h}_t = \tanh\left(W_c \cdot [h_t, c_t] + b_c\right) \tag{18}$$

where $\tilde{h}_t$ is the context-aware hidden state at time t, W_c is the learnable weight matrix for concatenation, h_t is the hidden state at time t, c_t is the context vector at time t, b_c is the bias term.

$$\beta_t = \sigma\left(W_b \cdot \tilde{h}_t + b_b\right) \tag{19}$$

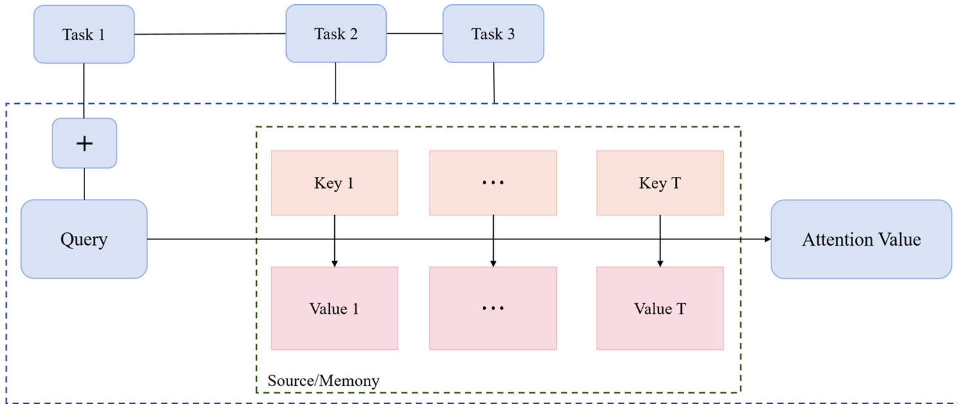

Fig. 4 Flow chart of the Dynamic-AM model

where β_t is the gating scalar at time t, W_b is the learnable weight matrix for the gating mechanism, $\tilde{h}_t$ is the context-aware hidden state at time t, b_b is the bias term.

$$h_{t+1} = \beta_t \odot \tilde{h}_t + (1 - \beta_t) \odot h_t \tag{20}$$

where h_{t+1} is the updated hidden state at time $(t + 1)$, β_t is the gating scalar at time t, $\tilde{h}_t$ is the context-aware hidden state at time t, h_t is the hidden state at time t.

3 Experiment

3.1 Experimental environment

- Hardware Environment
  The hardware environment used in the experiments consists of a high-performance computing server equipped with an AMD Ryzen Threadripper 3990X @ 3.70 GHz CPU and 1 TB RAM, along with 6 Nvidia GeForce RTX 3090 24 GB GPUs. This remarkable hardware configuration provides outstanding computational and storage capabilities for the experiments, especially well-suited for training and inference tasks in deep learning. It effectively accelerates the model training process, ensuring efficient experimentation and rapid convergence.
- Software Environment
  In this study, we utilized Python and PyTorch to implement our research work. Python, serving as the primary programming language, provided us with a flexible development environment. PyTorch, as the main deep learning framework, offered powerful tools for model construction and training. Leveraging PyTorch's computational capabilities and automatic differentiation functionality, we were able to efficiently develop, optimize, and train our models, thereby achieving better results in the experiments.

3.2 Experimental datasets

In order to study the performance of the ensemble temporal net model in smart pricing in the retail industry, we conducted a series of experiments. These experiments used several different data sets to comprehensively evaluate the model's performance in different sales scenarios to gain deeper insights.

The UCI Online Retail Data dataset is derived from online retail transaction data. This data set is rich in information and includes various sales-related variables, such as product characteristics, price, sales volume, etc. Meticulous recording of these dimensions allows us to gain a comprehensive understanding of the characteristics of retail transactions (John et al. 2023). In the experiment, we chose the UCI Online Retail Data dataset to evaluate the performance of the ensemble temporal net model in standard retail scenarios. Particular focus includes price optimization and sales forecasting, aiming to reveal the model's superiority in the actual sales process.

The Walmart Sales Forecasting data set is based on Walmart sales data and is a challenging competition data set (Wang and Gu 2022). This data set covers a large scale of real-world sales information, including product characteristics, price, sales volume and other dimensions. We deliberately selected this data set to test the generalization performance of the ensemble temporal net model in more complex and practical sales scenarios. By tackling this real-world challenge, we aim to gain a more complete understanding of model adaptability and robustness.

Online Retail II Data Set The data set originates from a British online retailer and contains detailed sales information. This dataset makes us ideal for studying the impact of different pricing strategies on sales. By digging deeper into this data set, we can gain a deeper understanding of the performance of the ensemble temporal net model in the context of different pricing strategies (Anitha and Patil 2022). We focus on how the model adjusts itself in the face of diverse sales strategies to provide more accurate sales forecasts.

The E-commerce Product Data dataset covers detailed information about products on e-commerce platforms. This data set is used to learn product features and sales prediction. The performance of the ensemble temporal net model on this data set will help us gain an in-depth understanding of the model's learning ability for product features. By testing its adaptability to different categories of products, we aim to ensure that the model maintains its predictive accuracy and robustness in the face of diverse product types (Cui et al. 2021).

3.3 Experimental setup and details

This research introduces the ensemble temporal net model and builds a deep learning-driven intelligent pricing model for the retail industry by integrating components of TFT, ES-RNN, and Dynamic-AM. To ensure the accuracy and reproducibility of experimental results, we will introduce the design and execution process of the experiment in detail.

Step 1: Dataset preparation

Data preprocessing plays a vital role in preparing raw data for our analysis and model development. In this section, we outline the key steps we take to ensure that our data is cleaned, normalized, and appropriately partitioned.

- Handle missing data: Identify and quantify missing values in your data set. If a feature has more than 5% missing values, technical imputation is performed.
- Data cleaning: Detect and handle outliers to mitigate their impact on model performance. Remove duplicate records to avoid redundancy and maintain data integrity. Resolve any inconsistencies or errors in the categorical data by standardizing categories or correcting spelling errors.
- Data normalization: Standardize numerical features so that they have a mean of 0 and a standard deviation of 1. Apply min-max scaling to ensure features are within a specific range, usually between 0 and 1. Categorical variables are encoded using one-hot encoding for compatibility with certain machine learning algorithms.
- Data splitting: Split the data set into a training set and a test set to evaluate the generalization performance of the model. The training and testing split ratios are 70–30 or 80–20 respectively.

Step 2: Model training

- Network parameter settings: For the feedforward neural network part, two hidden layers were selected, each containing 128 neurons. The learning rate can be set to 0.005. For the CNN and RNN components, set up two convolutional layers with the number of filters 64, and an RNN layer with 32 memory units.
- Model architecture design: TFT component, using two Transformer Encoder layers, each layer has 4 heads, the hidden layer size of each head is 32, and Positional Encoding is used to capture time information. The ES-RNN component integrates 3 simplified RNN models, each RNN model contains 64 memory units. The Dynamic-AM component assigns dynamic attention weights to different time points or products based on the dynamics of sales data.
- Model training process: After determining the network parameters and architecture of the integrated model, the actual training of the model is carried out. Choose the number of iterations as 80 and the batch size as 64. During training, monitor performance metrics on the training and validation sets. Stochastic gradient descent was used as the optimization algorithm, and the learning rate decay was set to 0.95.

Step 3: Model validation and tuning

- Cross-Validation: In order to accurately evaluate the generalization performance of the model, this article uses k-fold cross-validation (k-Fold Cross-Validation). The data set is divided into 5 subsets, 4 subsets are used for training each time, and one subset is reserved for verification. This process is repeated 5 times to ensure that each subset is used as a validation set. Finally, the performance of each verification is calculated and averaged as the final performance evaluation index.
- Model Fine-Tuning: Based on cross-validation, perform model tuning to further improve performance. The main parameters to be adjusted include the learning rate of the integrated TFT, the number of hidden units of ES-RNN, and the convolution kernel size of the CNN. We varied the learning rate from 0.0001 to 0.01, varied the number of hidden units from 64 to 256, and tried different sizes of convolution kernels. Ultimately, the parameter combination with the best performance in cross-validation is selected to ensure that the deep learning model achieves the best performance on the given data set.

Step 4: Ablation experiment

In the experimental design of this paper, we conduct a series of ablation experiments to deeply study the impact of various components of the deep learning-driven retail intelligent pricing model on model performance.

- Removing Attention Mechanism: The Dynamic-AM component is removed from the model. The model parameters are initialized randomly and optimized solely using the standard gradient descent method. The learning rate is set to 0.001, batch size to 64, and the number of training iterations is fixed at 50 epochs.

- Removing ES-RNN: The ES-RNN component is removed from the model, and the outputs from the Temporal Fusion Transformer and Dynamic Attention Mechanism are directly connected to the output layer. To maintain the complexity of the model, the number of neurons in the output layer is increased. The learning rate is set to 0.001, batch size to 64, and the training iterations are fixed at 50 epochs.
- Removing Temporal Fusion Transformer: The TFTr component is removed from the model, which then starts processing features directly from the input layer. The number and architecture of the fully connected layers remain unchanged to ensure that the model retains a certain level of complexity. The input dimension of the first fully connected layer is adjusted to match the feature dimension of the original output from the Temporal Fusion Transformer. The model's learning rate is set to 0.001, batch size to 64, and training iterations are fixed at 50 epochs.

Step 5: Comparative Analysis

This article also conducted a series of ablation experiments, focusing on optimization strategies, to compare the performance between Self-Attention Mechanism (Self-AM), Multi-Head Attention Mechanism (Multi-Head-AM), Cross Attention Mechanism (Cross-AM), and Dynamic Attention Mechanism (Dynamic-AM).

- Self-AM versus Dynamic-AM: Firstly, we compared the performance of Self-AM and Dynamic-AM in sales forecasting at different price levels. The model's learning rate was set to 0.001, batch size to 64, and the number of training iterations to 50 epochs.
- Multi-Head-AM versus Dynamic-AM: We analyzed and compared the performance between Multi-Head-AM and Dynamic-AM. The model's learning rate was set to 0.001, batch size to 64, and the training iterations to 50 epochs.
- Cross-AM versus Dynamic-AM: A detailed comparison was carried out between Cross-AM and Dynamic-AM. We maintained the model's learning rate at 0.001, batch size at 64, and the number of training epochs at 50.

Step 6: Model Evaluation

In this study, we conduct a comprehensive evaluation of the model, focusing mainly on its accuracy and efficiency in price optimization and sales forecasting. The following are some performance indicators:

- Accuracy evaluation indicators: In order to evaluate the accuracy of the model, we use a number of commonly used indicators, including: MAE, MAPE, RMSE, and MSE.
- Efficiency evaluation metrics: In order to evaluate the efficiency of the model, we considered the following metrics: Parameters, Flops, Inference Time, and Training Time.

4 Results

As shown in Table 1, we conduct a comprehensive performance comparison with other methods on four datasets. Our model performs better in various indicators than methods such as Lu, Zhang, Jing, Singh, Khalil, and Mishra. On UCI Online Retail Data, compared to the Lu method, our model reduces MAE by 46.4%, MAPE by 63.65%, RMSE by 54.78%, and MSE by 17.71%. On other data sets, our method also achieved similar advantages and showed higher accuracy. Especially on E-commerce Product Data, compared to the Khalil method, the MAE of our model is reduced by 59.72%, the MAPE is reduced by 30.41%, the RMSE is reduced by 36.87%, and the MSE is reduced by 35.09%. Experimental results show that our model is more suitable for intelligent pricing tasks, can predict sales more accurately, and provide more optimized pricing decisions for the retail industry. Figure 5 visualizes the contents of the table and more clearly demonstrates the superiority of our model over other methods (Table 2).

As shown in Tables 3, 4, 5 and 6, we conducted a comprehensive performance comparison with other methods on four datasets, focusing on indicators such as the number of parameters, computational complexity, inference time, and training time. Our model performs better in various indicators than methods such as Lu, Zhang, Jing, Singh, Khalil, and Mishra. In terms of the number of parameters, our model is reduced by 34.07% compared to the Lu method, 57.75% compared to the Zhang method, 42.42% compared to the Jing method, 49.32% compared to the Singh method, and reduced by 49.32% compared to the Khalil method. 29.27%, which is a reduction of 0.22% compared to the Mishra method. This shows that our model has a more streamlined network structure while maintaining effectiveness. In terms of computational complexity (Flops), our model is reduced by 43.71% compared to the Lu method, 59.30% compared to the Zhang method, 54.33% compared to the Jing method, 45.78% compared to the Singh method, and 45.78% compared to the Singh method. The Khalil method reduced it by 33.15%, which was 4.55% compared to the Mishra method. This shows that our model is more efficient in utilizing computing resources. In terms of inference time and training time, our model is significantly reduced compared to methods such as Lu, Zhang, Jing, Singh, Khalil, and Mishra. For example, on UCI Online Retail Data, our model reduces inference time by 41.84% and training time by 43.85% compared to the Lu method. On other data sets, our method has achieved similar advantages. Figure 6 visualizes the contents of the table and

Table 1 The comparison of different models in different indicators comes from different Datasets (Part 1)

Model	UCI online retail data				Walmart sales forecasting			
	MAE	MAPE	RMSE	MSE	MAE	MAPE	RMSE	MSE
Lu et al. (2020)	34.8	12.49	8.16	14.03	40.23	14.98	6.97	13.34
Zhang et al. (2019)	33.77	14.51	7.68	21.44	41.74	11.27	8.21	29.23
Jing et al. (2021)	29.29	8.7	7.02	17.79	48.85	15.16	6.69	24.78
Singh et al. (2020)	43.32	12.19	5.66	20.97	27.45	11.64	4.98	13.23
Khalil et al. (2021)	32.35	12.01	6.22	15.68	31.05	10.75	4.77	21.14
Mishra and Tyagi (2022)	20.74	10.81	6	15.33	48.04	13.94	5.69	22.24
Ours	18.66	4.53	3.85	11.52	17.83	4.95	3.03	10.65

Table 2 The comparison of different models in different indicators comes from different Datasets (Part 2)

Model	Online retail II data set				E-commerce product data			
	MAE	MAPE	RMSE	MSE	MAE	MAPE	RMSE	MSE
Lu	20.88	11.47	5.79	20.31	28.16	14.27	5.78	17.65
Zhang	25.16	8.59	5.24	26.23	48.52	13.33	6.08	30.34
Jing	45.83	10.68	5.53	13.53	42.69	9.04	8.47	22.94
Singh	38.42	9.35	5.01	22.94	35.92	9.78	5.59	22.05
Khalil	46.15	8.65	4.54	12.44	32.76	14.85	7.56	16.73
Mishra	45.6	11.06	5.08	23.8	25.83	11.14	4.8	28.9
Ours	18.2	6.14	4.15	10.86	15.89	5.52	3.81	10.92

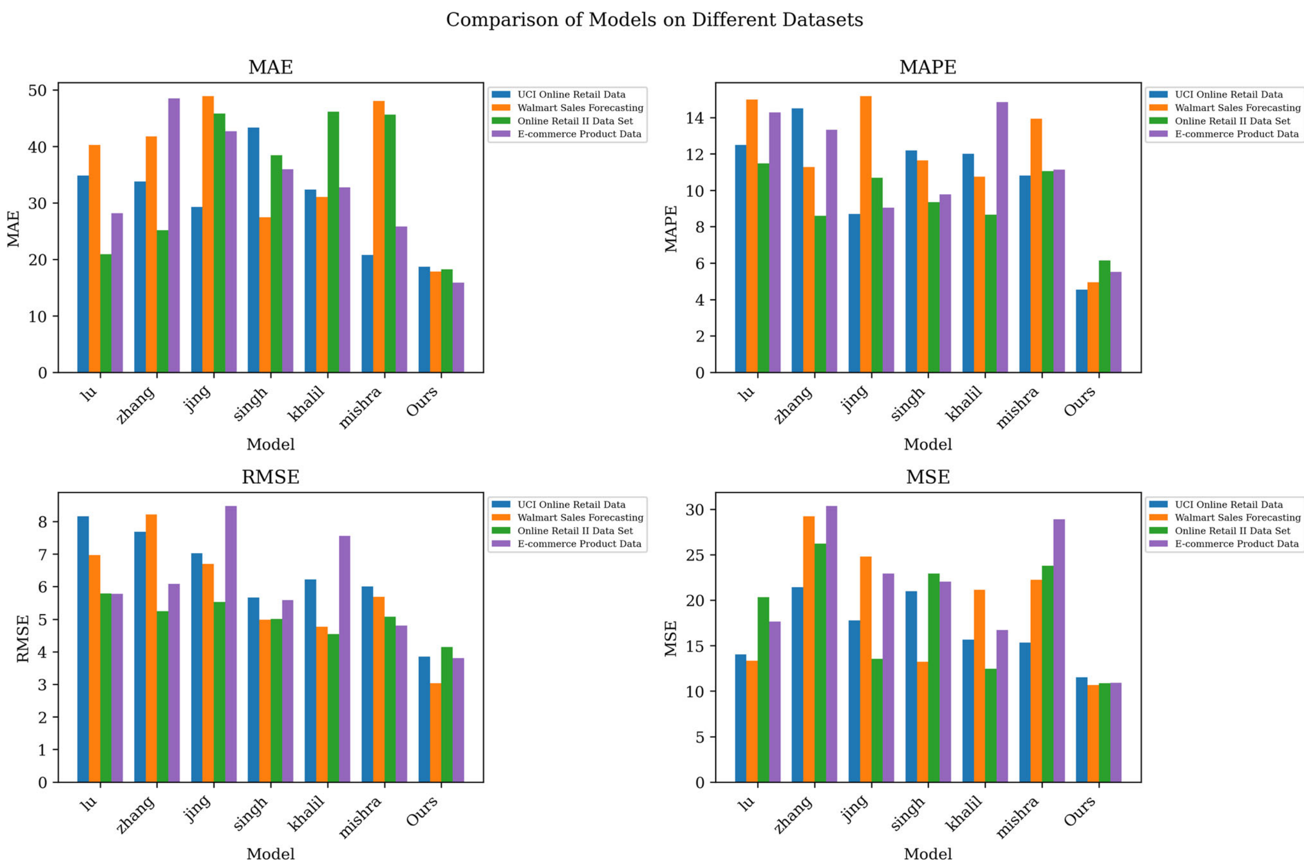

Fig. 5 Model accuracy verification comparison chart of different indicators of different models

Table 3 Model efficiency verification and comparison of different indicators of different Datasets (Part 1)

Model	UCI online retail data			
	Parameters (M)	Flops (G)	Inference time (ms)	Training time (s)
Lu	510.98	6.26	9.19	580.31
Zhang	798.15	8.64	10.22	789.58
Jing	522.44	7.70	6.69	729.32
Singh	664.34	6.46	11.67	708.03
Khalil	475.59	5.17	7.61	414.27
Mishra	337.06	3.52	5.37	327.49
Ours	336.79	3.52	5.34	326.14

Table 4 Model efficiency verification and comparison of different indicators of different Datasets (Part 2)

Model	Walmart sales forecasting			
	Parameters (M)	Flops (G)	Inference time (ms)	Training time (s)
Lu	545.10	5.34	8.86	540.62
Zhang	607.47	8.48	12.84	777.01
Jing	592.83	7.76	12.44	674.60
Singh	700.38	6.92	12.22	767.28
Khalil	390.76	4.93	7.17	421.08
Mishra	318.21	3.65	5.59	335.79
Ours	317.15	3.65	5.63	335.28

Table 5 Model efficiency verification and comparison of different indicators of different Datasets (Part 3)

Model	Online retail II data set			
	Parameters (M)	Flops (G)	Inference time (ms)	Training time (s)
Lu	514.69	5.11	9.35	573.03
Zhang	803.27	7.29	12.65	773.65
Jing	797.16	7.84	11.94	775.95
Singh	680.71	8.22	9.68	693.01
Khalil	420.31	4.51	7.42	410.76
Mishra	339.51	3.53	5.34	327.82
Ours	337.70	3.63	5.64	326.73

Table 6 Model efficiency verification and comparison of different indicators of different Datasets (Part 4)

Model	E-commerce product data			
	Parameters (M)	Flops (G)	Inference time (ms)	Training time (s)
Lu	446.36	6.50	9.60	528.98
Zhang	639.66	8.90	12.47	651.93
Jing	493.10	7.32	10.55	542.49
Singh	694.66	7.48	12.80	649.57
Khalil	437.78	5.28	8.39	427.30
Mishra	317.92	3.65	5.60	337.79
Ours	318.47	3.64	5.64	338.23

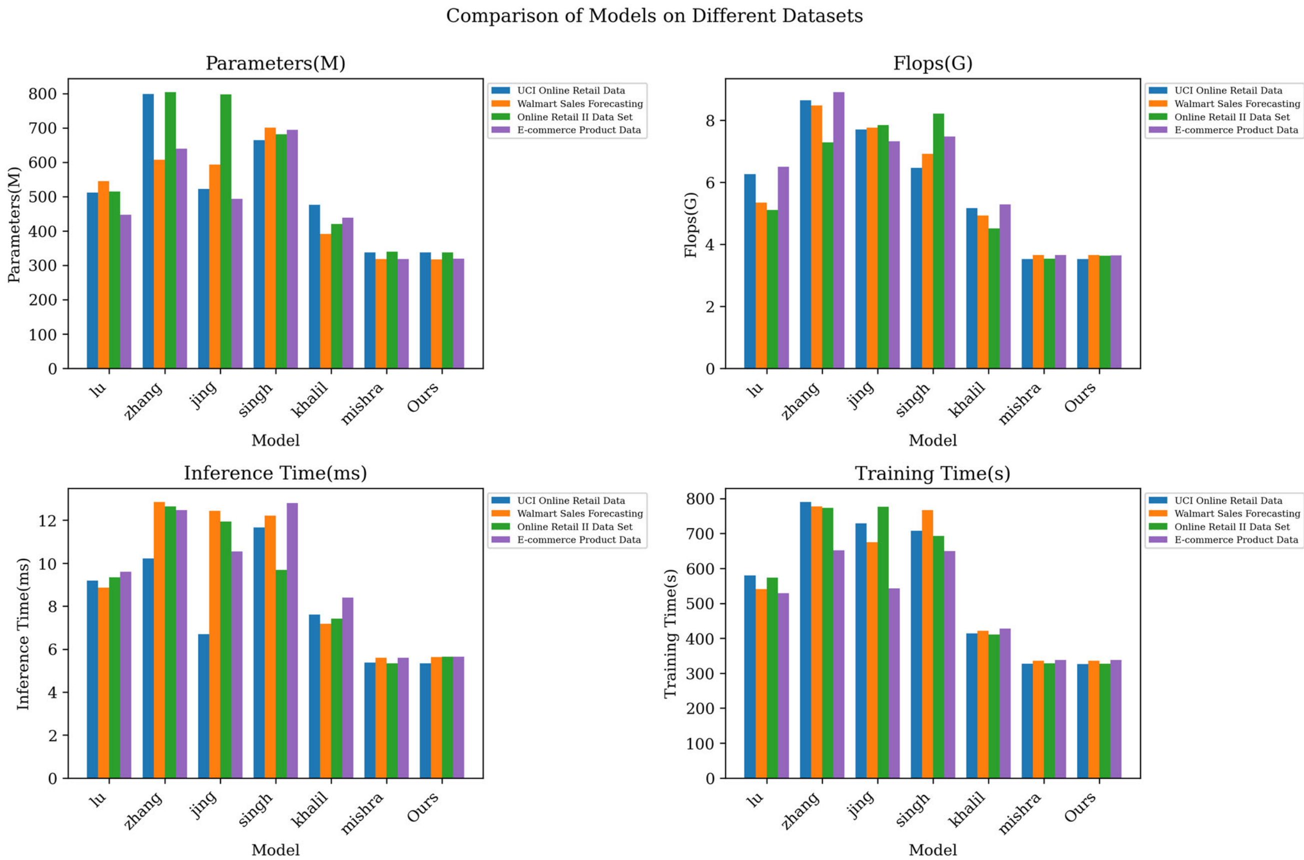

Fig. 6 Model efficiency verification comparison chart of different indicators of different models

more clearly demonstrates the superiority of our model over other methods.

As shown in Tables 7 and 8, we conducted a series of ablation experiments to deeply study the impact of each component of the model on the overall performance by comparing the performance indicators of different model configurations on various data sets. First of all, for the case where the attention mechanism is not used, compared with the TFT+ES-RNN model, removing the Dynamic Attention Mechanism component leads to a significant decrease in performance indicators. On the E-commerce Product Data dataset, the MAE increases from 22.60 to 25.60 and the RMSE increases from 6.58 to 7.14, which shows that the dynamic attention mechanism is crucial to improve model performance when processing multi-variable time series data. Secondly, for the case where ES-RNN is not used, the ES-RNN+Dynamic AM model is used. Compared with the TFT+ES-RNN model, the performance is improved on the two data sets of Walmart Sales Forecasting and Online Retail II Data Set, but dropped slightly on the other two datasets. This shows that ES-RNN has a positive effect on improving model performance to a certain extent, but its effects on different data sets are different. Finally, for the case where the Temporal Fusion Transformer is not used, the removal of the Temporal Fusion Transformer leads to a significant decrease in performance on all datasets compared to the TFT+ES-RNN model. For example, on the E-commerce Product Data dataset, MAE increases from 25.48 to 38.25 and RMSE increases from 5.43 to 6.56. The effectiveness of Temporal Fusion Transformer in multi-level time series data processing has been verified. Comprehensive comparison shows that our comprehensive model achieves the best performance in all indicators. Figure 7 visualizes the contents of the table and more clearly demonstrates the superiority of our comprehensive model in various indicators.

As shown in Tables 9, 10, 11 and 12, we conducted a series of comparative experiments to compare the performance of four different attention mechanisms: Self-AM, Multi-Head-AM, Cross-AM and Dynamic-AM. First, comparing Self-AM and Dynamic-AM, we observe that Dynamic-AM is significantly better than Self-AM in terms of number of parameters, calculation amount, inference time and training time on all data sets. For example, on the E-commerce Product Data dataset, the number of parameters of Dynamic-AM is 207.79M, while the number of parameters of Self-AM is 375.33M, indicating that Dynamic-AM achieves better results while maintaining a relatively low number of parameters. good performance. Similar advantages are reflected in

Table 7 Ablation experiments on the Ensemble TemporalNet module using different datasets (Part 1)

Model	UCI online retail data				Walmart sales forecasting			
	MAE	MAPE	RMSE	MSE	MAE	MAPE	RMSE	MSE
TFT+ES-RNN	21.53	12.12	4.95	12.36	22.62	15.16	7.75	17.50
ES-RNN+Dynamic AM	25.60	12.85	7.14	28.77	23.12	8.97	6.23	22.76
TFT+ES-RNN	27.71	11.37	4.23	18.66	33.12	13.81	6.58	25.27
Ours	16.02	6.39	3.49	8.85	12.05	8.16	3.33	8.06

Table 8 Ablation experiments on the Ensemble TemporalNet module using different datasets (Part 2)

Model	Online retail II data set				E-commerce product data			
	MAE	MAPE	RMSE	MSE	MAE	MAPE	RMSE	MSE
TFT+ES-RNN	38.25	12.65	6.56	29.62	37.37	12.58	4.79	20.85
ES-RNN+Dynamic AM	30.84	10.41	5.22	13.41	22.60	13.54	5.62	26.41
TFT+ES-RNN	25.48	11.17	5.43	23.15	41.37	11.72	7.69	20.73
Ours	19.64	5.85	2.47	9.27	12.67	7.03	3.35	10.80

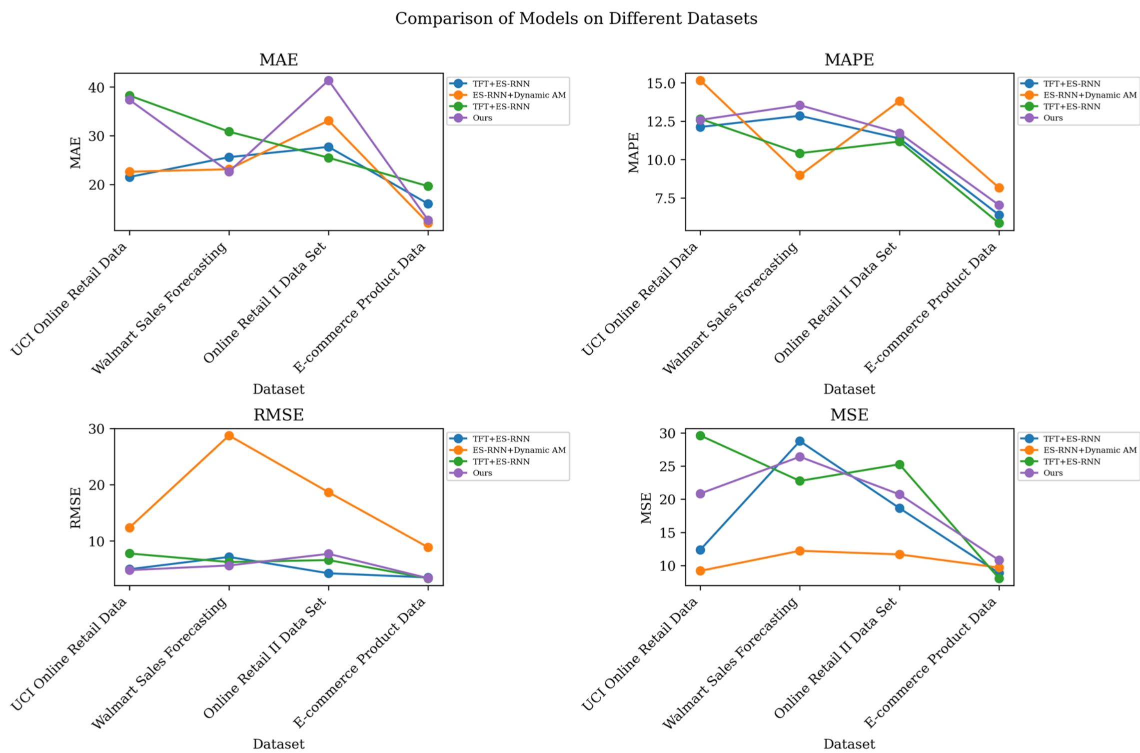

Fig. 7 Ablation experiments on the Ensemble TemporalNet module

other indicators. Secondly, comparing Multi-Head-AM and Dynamic-AM, we find that Dynamic-AM shows better performance on all data sets. Taking Walmart Sales Forecasting as an example, the MAE of Dynamic-AM is 159.96, while the MAE of Multi-Head-AM is 277.42. Dynamic-AM is more accurate in sales forecasting. Finally, comparing Cross-AM and Dynamic-AM, Cross-AM achieved better performance on some data sets, but when combining various indicators, Dynamic-AM is still a superior choice. Taking Online Retail II Data Set as an example, Dynamic-AM has achieved better performance in MAE, MAPE, RMSE and MSE. Overall, Dynamic-AM achieved relatively good performance on different data sets, proving its effectiveness in intelligent pricing models in the retail industry. Figure 8 visualizes the contents of the table and more clearly demonstrates the obvious advantages of Dynamic-AM over other attention mechanisms in different indicators.

Table 9 Ablation experiments on the Dynamic AM using different datasets (Part 1)

Model	UCI online retail data			
	Parameters (M)	Flops (G)	Inference time (ms)	Training time (s)
Self-AM	375.33	272.86	254.29	310.38
Multi-Head-AM	397.46	304.81	269.29	288.55
Cross-AM	333.34	369.39	266.71	310.47
Dynamic-AM	207.79	179.21	205.23	229.26

Table 10 Ablation experiments on the Dynamic AM using different datasets (Part 2)

Model	Walmart sales forecasting			
	Parameters (M)	Flops (G)	Inference time (ms)	Training time (s)
Self-AM	365	381.97	214.58	416.76
Multi-Head-AM	277.42	356.34	393.04	347.02
Cross-AM	355.24	337.53	266.9	364.44
Dynamic-AM	159.96	181.89	196.91	107.59

Table 11 Ablation experiments on the Dynamic AM using different datasets (Part 3)

Model	Online retail II data set			
	Parameters (M)	Flops (G)	Inference time (ms)	Training time (s)
Self-AM	375.36	302.09	291.82	384.28
Multi-Head-AM	375.06	271.85	246.48	292.6
Cross-AM	305.59	315.18	235.11	284.31
Dynamic-AM	158.87	114.57	232.92	187.83

5 Conclusion

In the fierce competition and ever-changing market environment in the digital era, this study is dedicated to solving the challenges faced by the retail industry in pricing strategies. By introducing deep learning technology, we built an intelligent pricing model that integrates TFT, ES-RNN, and Dynamic-AM. We conduct extensive experiments on four different datasets, demonstrating the model's superior performance in sales prediction and price optimization. This comprehensive model not only adapts to the complexity and dynamics of the market, but also shows obvious advantages in improving business efficiency. Experimental results show that our model has certain advantages and can be well adapted to intelligent pricing tasks, thereby predicting sales more accurately and providing more optimized pricing decisions for the retail industry.

However, we are also aware of some potential shortcomings of the model. First, the model may perform unstable when dealing with extreme market fluctuations, and more research on adversarial training is needed. Secondly, for certain product-specific sales scenarios, there is still room for improvement in the model's generalization capabilities, which requires the support of more domain-specific data and feature engineering.

This paper experimentally verifies the feasibility and effectiveness of a deep learning-driven intelligent pricing model for the retail industry, providing retailers with a more flexible and intelligent pricing strategy. In addition, in actual business, this model can not only predict sales more accurately, but also adjust pricing in real time to adapt to market changes. The research in this article promotes the intelligent development of pricing strategies in the retail industry and provides enterprises with more competitive tools. In

Table 12 Ablation experiments on the Dynamic AM using different datasets (Part 4)

Model	E-commerce product data			
	Parameters (M)	Flops (G)	Inference time (ms)	Training time (s)
Self-AM	277.58	256.27	331.46	389.47
Multi-Head-AM	379.96	292.47	220.43	401.62
Cross-AM	356.64	285.89	389.75	391.95
Dynamic-AM	213.86	220.18	200.68	195.26

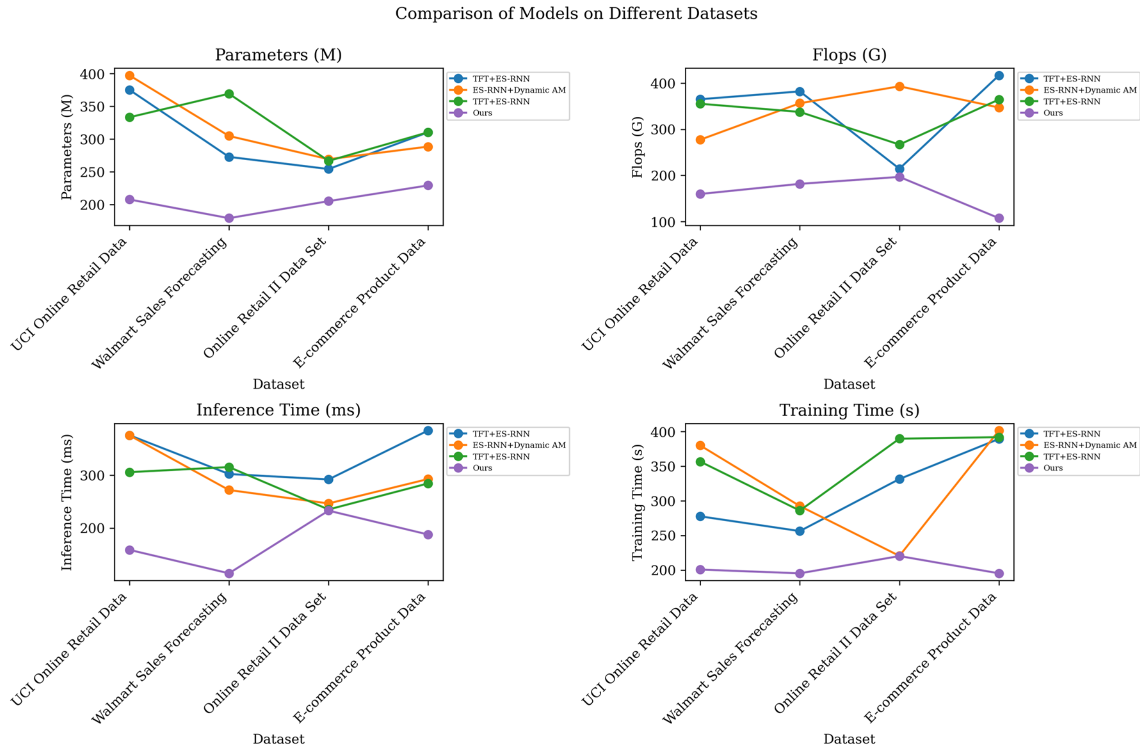

Fig. 8 Ablation experiments on the Dynamic AM

the future, we will further improve the model, strengthen its adaptability to extreme scenarios, and conduct in-depth research on applications in more fields, such as promotion strategies and inventory management, to comprehensively improve the level of intelligence in the retail industry.

Author contributions Dongxin Li and Jiayue Xin contributed to the conception and design of this study. Dongxin Li was solely responsible for model construction and data collection. Jiayue Xin was responsible for statistical analysis and wrote the first draft.

Funding This work was supported by the general project of the national social science foundation of China (CNSSF) in 2018, named "Research on the Construction of New East Asian Economic Cooperation Model under the Background of 'One Belt One Road' (No. 18BGJ001); Also supported by Shandong University (Weihai) Young Scholars' Future Plan Project named" Research on the Construction of New East Asian Economic Cooperation Model in the New Situation" (No. 20820201007).

Data availability Not applicable.

Code availability Not applicable.

Declarations

Conflict of interest The 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.

Consent to participate Not applicable.

Consent for publication Not applicable.

Ethical approval Not applicable.

References

Anitha P, Patil MM (2022) RFM model for customer purchase behavior using k-means algorithm. J King Saud Univ Comput Inform Sci 34(5):1785–1792

Arora P, Balyan A (2023) Comparative analysis of LSTM, encoder-decoder and GRU models for stock price prediction. In: Computational intelligence: select proceedings of InCITe 2022. Springer, pp 399–410

Bao T, Ren N, Luo R, Wang B, Shen G, Guo T (2021) A Bert-based hybrid short text classification model incorporating CNN and attention-based Bigru. J Organ End User Comput (JOEUC) 33(6):1–21

Chen G, Zhu D, Chen X (2023) Similarity detection method of science fiction painting based on multi-strategy improved sparrow search algorithm and gaussian pyramid. Multimed Tools Appl 2023:1–40

Cui F, Hu H, Xie Y (2021) An intelligent optimization method of e-commerce product marketing. Neural Comput Appl 33:4097–4110

Du Y, Li F (2019) Intelligent multi-microgrid energy management based on deep neural network and model-free reinforcement learning. IEEE Trans Smart Grid 11(2):1066–1076

Falatouri T, Darbanian F, Brandtner P, Udokwu C (2022) Predictive analytics for demand forecasting-a comparison of Sarima and LSTM in retail SCM. Proc Comput Sci 200:993–1003

Gao T, Wang C, Zheng J, Wu G, Ning X, Bai X, Yang J, Wang J (2023) A smoothing group lasso based interval type-2 fuzzy neural network for simultaneous feature selection and system identification. Knowl Based Syst 280:111028

Godahewa R, Bandara K, Webb GI, Smyl S, Bergmeir C (2021) Ensembles of localised models for time series forecasting. Knowl Based Syst 233:107518

Gupta S, Maulik A, Das D, Singh A (2022) Coordinated stochastic optimal energy management of grid-connected microgrids considering demand response, plug-in hybrid electric vehicles, and smart transformers. Renew Sustain Energy Rev 155:111861

Jing N, Wu Z, Wang H (2021) A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction. Expert Syst Appl 178:115019

John JM, Shobayo O, Ogunleye B (2023) An exploration of clustering algorithms for customer segmentation in the UK retail market. Analytics 2(4):809–823

Khalid R, Javaid N, Al-Zahrani FA, Aurangzeb K, Qazi E-u-H, Ashfaq T et al (2020) Electricity load and price forecasting using Jaya-long short term memory (JLSTM) in smart grids. Entropy 22(1):10

Khalil RA, Saeed N, Masood M, Fard YM, Alouini M-S, Al-Naffouri TY (2021) Deep learning in the industrial internet of things: potentials, challenges, and emerging applications. IEEE Int Things J 8(14):11016–11040

Li K, Shen N, Kang Y, Chen H, Wang Y, He S (2021) Livestock product price forecasting method based on heterogeneous GRU neural network and energy decomposition. IEEE Access 9:158322–158330

Liang F, Zhang H, Fang Y (2022) The analysis of global RMB exchange rate forecasting and risk early warning using ARIMA and CNN model. J Organ End User Comput (JOEUC) 34(8):1–25

Lu T, Chen X, McElroy MB, Nielsen CP, Wu Q, Ai Q (2020) A reinforcement learning-based decision system for electricity pricing plan selection by smart grid end users. IEEE Trans Smart Grid 12(3):2176–2187

Metin A, Kasif A, Catal C (2023) Temporal fusion transformer-based prediction in aquaponics. J Supercomput 2023:1–25

Mishra S, Tyagi AK (2022) The role of machine learning techniques in internet of things-based cloud applications. In: Artificial intelligence-based internet of things systems, pp 105–135

Mujeeb S, Javaid N, Ilahi M, Wadud Z, Ishmanov F, Afzal MK (2019) Deep long short-term memory: a new price and load forecasting scheme for big data in smart cities. Sustainability 11(4):987

Ruan J, Liu G, Qiu J, Liang G, Zhao J, He B, Wen F (2022) Time-varying price elasticity of demand estimation for demand-side smart dynamic pricing. Appl Energy 322:119520

Semenoglou A-A, Spiliotis E, Assimakopoulos V (2023) Image-based time series forecasting: a deep convolutional neural network approach. Neural Netw 157:39–53

Shao W, Sun H, Wang Y, Qing X (2022) A multi-level damage classification technique of aircraft plate structures using lamb wave-based deep transfer learning network. Smart Mater Struct 31(7):075019

Singh SK, Jeong Y-S, Park JH (2020) A deep learning-based IoT-oriented infrastructure for secure smart city. Sustain Cities Soc 60:102252

Soleimani M, Kezunovic M (2020) Mitigating transformer loss of life and reducing the hazard of failure by the smart EV charging. IEEE Trans Ind Appl 56(5):5974–5983

Sujith A, Sajja GS, Mahalakshmi V, Nuhmani S, Prasanalakshmi B (2022) Systematic review of smart health monitoring using deep learning and artificial intelligence. Neurosci Inform 2(3):100028

Sunny MAI, Maswood MMS, Alharbi AG (2020) Deep learning-based stock price prediction using LSTM and bi-directional LSTM model. In: 2020 2nd novel intelligent and leading emerging sciences conference (NILES). IEEE, pp 87–92

Theodoridis G, Tsadiras A (2022) Comparing boosting and deep learning methods on multivariate time series for retail demand forecasting. In: IFIP international conference on artificial intelligence applications and innovations. Springer, London, pp 375–386

Wang C-H, Gu Y-W (2022) Sales forecasting, market analysis, and performance assessment for US retail firms: a business analytics perspective. Appl Sci 12(17):8480

Wang P-Y, Chen C-T, Su J-W, Wang T-Y, Huang S-H (2021) Deep learning model for house price prediction using heterogeneous data analysis along with joint self-attention mechanism. IEEE Access 9:55244–55259

Xiao X, Xu W, Wang H, Su P, Gao S (2021) Deep face representation algorithm based on dual attention mechanism. J Jilin Univ Sci Ed 59(4):883–890

Xiao X, Xu W, Wang H, Su P, Gao S (2022) Chinese caption of fine-grained images based on transformer. J Jilin Univ Sci Ed 60(5):1103–1112

Zhang Z, Zhang D, Qiu RC (2019) Deep reinforcement learning for power system applications: an overview. CSEE J Power Energy Syst 6(1):213–225