Journal of Engineering, Project, and Production Management
2026, 16(5), 2026-45

An Intelligent Prediction Model for Accounting Cost Calculation Based on Temporal Convolution and Attention Mechanisms

Tong Wu[1] and Shuang Wu[2]

[1] Undergraduate Student, School of Accounting, Harbin University of Commerce, Harbin, 150000, China, E-mail: wutong521222@outlook.com (corresponding author).

[2] Undergraduate Student, Business School, University of New South Wales, Sydney, 2033, Australia, E-mail: wushuang846034@163.com

Project Management
Received January 13, 2026; revised March 12, 2026; accepted July 16, 2026
Available online July 25, 2026

Abstract: Accounting cost series exhibit multi-scale fluctuations and sudden disturbances, making it difficult for traditional statistical models and single deep structures to achieve stable predictive performance. To improve the accuracy and robustness of accounting cost prediction, this paper proposes an intelligent model that integrates multi-scale temporal convolution and channel-temporal collaborative attention. This model constructs stable and learnable input segments through normalization, anomaly suppression, and a bidirectional sliding window, and captures multi-granularity dependencies using parallel dilated convolution. Attention is used to dynamically weight key cost elements and key time points. Experimental results show that the proposed model outperforms the comparative models in prediction accuracy, with mean absolute error, root mean square error, and mean absolute percentage error of 0.07, 0.12, and 8.50%, respectively. In multi-product line prediction, the mean absolute error stabilizes between 0.07 and 0.09, and the root mean square error is below 0.12. In the presence of noise and missing data, the root mean square error remains between 0.08 and 0.09, demonstrating the best performance in bias control. The mean absolute error for the four prediction periods is 0.30%, and the inference latency is stable at approximately 2.0ms to 2.8ms. The proposed forecast model provides effective support for enterprise cost control and rolling budget. The research satisfies the dual requirements of forecast excellence and proven decision impact, providing a robust mechanism for converting fine-grained accounting data into actionable financial intelligence, thereby minimizing structural budget waste and maximizing the strategic utility of rolling forecasts in complex business environments.

Keywords: Accounting cost forecasting; Multi-scale temporal convolution; Attention mechanism; Intelligent financial data management; Rolling budget

DOI 10.32738/JEPPM-2026-45

---

1. Introduction

Against the backdrop of digitalization, accounting cost prediction directly affects budget preparation, resource allocation and cost control (Wallis, 2023; Boone and Yin, 2024). However, accounting cost series are often affected by interactions between external and internal factors, such as purchase price fluctuations and accounting delays (Ahabchane et al., 2024; Petropoulos et al., 2025). Traditional statistical models are prone to systematic bias when facing complex dynamics (Aydinli, 2024; Patel et al., 2024). Although deep learning methods have made breakthroughs in nonlinear modeling, most rely solely on single-scale convolution or sequence encoding, which is insufficient to capture multi-period coupled fluctuations. Moreover, in the context of parallel changes in multiple cost factors, they lack the ability to assign differentiated weights to key indicator channels and key time-series points (Chen et al., 2022; Yan et al., 2024). To address these issues, academia and industry are gradually exploring the integration of temporal convolution, attention mechanisms, and business-scenario modeling to enhance the predictive model's ability to capture local fluctuations and asynchronous dependencies.

To address the high uncertainty in project cost prediction, Ünsal-Altuncan and Vanhoucke (2024) proposed a hybrid prediction model based on Bayesian networks that infers cost-schedule deviations using causal structures and integrates historical performance data. Experiments showed that the prediction error was significantly lower than that of traditional regression methods, and the robustness was stronger. To improve the reliability of engineering cost estimation, Narbaev et al. (2024) used a variety of machine learning algorithms to develop an automated cost deviation correction mechanism and to reduce systematic overestimation/underestimation errors through feature importance screening and cross-validation. Experiments showed that the prediction reliability was significantly improved. To address the non-stationarity and multi-scale fluctuations in the construction cost index, Wang et al. (2025) combined variational mode decomposition, long short-term memory networks, and gated recurrent units to construct a hybrid model that first decomposes trends and noise, then learns time-series dependencies hierarchically. Experimental results showed that the proposed model outperformed statistical and single-deep models. To address the highly nonlinear problem of financial time series, Dossatayev et al. (2023) proposed a deep prediction structure that integrates residual convolutional networks and long short-term memory networks, which enhances multi-scale feature expression and retains long-term time series memory through residual enhancement. Experimental results verified the effectiveness of the method. In response to the dynamic risk constraint problem in the cost optimization of green bonds, Hu et al. (2024) constructed a multi-stage prediction model and achieved optimal control of green bond issuance costs through hierarchical risk constraints and a dynamic backtracking mechanism, effectively improving prediction stability and economic adaptability. In the context of Internet finance and the digital economy, Huang (2023) used machine learning algorithms to model consumer finance data, enable collaborative prediction of financial risk and consumer behavior, and provide data support and methodological references for the digital transformation of the financial market. In response to the problem of corporate capital cost calculation, Olson and Pagano (2024) et al. proposed an empirical method for estimating the average cost of capital, quantifying capital costs at the corporate and industry levels, and providing empirical evidence for financial decision-making and capital structure optimization.

While the aforementioned studies have made progress in improving the accuracy of cost prediction, they mostly employ decomposition and single-scale modeling frameworks, which make it difficult to simultaneously capture short-term sudden fluctuations and medium- to long-term trends. Explicit modeling of channel importance and key-point contributions across different cost elements is lacking, and robustness verification against complex business disturbances, such as raw material price shocks and delayed accounting cycles, is insufficient. Furthermore, inference performance fails to meet the requirements of real-time rolling budgets. Therefore, to improve the accuracy and robustness of accounting cost prediction, this study proposes the multi-scale temporal convolution network with channel-temporal collaborative attention mechanism (MTCN-CAM). This model enhances the expression of key financial features through dual-dimensional attention and maintains low error and millisecond-level inference performance in noisy and missing environments, thus overcoming the shortcomings of existing research in robustness, bias control, and engineering deployment efficiency. Theoretical Framework: This study sits within the "Information-Decision" value loop. Accurate forecasting via MTCN-CAM provides the data foundation, which reduces budget slack in rolling budgeting systems. This alignment enables managers to transition from reactive cost accounting to proactive resource allocation and early warning of risks. The main innovations of the research are: (1) MTCN with parallel expansion branches and bottleneck residual fusion can simultaneously capture daily peaks and quarterly cycles; (2) CAM extracts inter-channel dependencies by applying global average pooling before calculating temporal weights. CAM can effectively filter accounting noise and highlight key cost drivers, achieving higher accuracy and lower latency than globally focused TFT. This research provides a deployable and scalable intelligent solution for enterprise cost forecasting, budget management, and cost risk early warning.

2. Methods and Materials

2.1. Analysis and Preprocessing Methods for Accounting Cost Series Data

The cost formation process of enterprises involves multiple links, such as procurement, production, labor, and energy. The efficiency of collaboration, information transmission mechanism, and performance evaluation system of these links will affect the authenticity and timeliness of cost data. The behavioral choices of different departments in balancing interests, allocating budgets, and competing for resources often lead to structural deviations and periodic fluctuations in the data. Meanwhile, the external policy environment, market price changes, and supply chain relationships will also affect the rhythm and structure of cost recording through institutional transmission mechanisms. Accounting cost data, therefore, has typical social embedding. Its time series changes not only reflect the economic cycle of enterprise production and operation, but also carry the social significance of managers' decision-making preferences, organizational coordination capabilities, and governance levels. As a typical financial time series, accounting cost data has statistical characteristics such as seasonality and multi-scale volatility (Ghadimi and Powell, 2024; Easton et al., 2024). To improve the predictability of cost indicators and the efficiency of model learning, a cost preprocessing mechanism was constructed. This mechanism maps the original cost records to a stable learning space containing local contextual semantics. The original cost sequence is normalized to eliminate the impact of different cost scales on the model. The cost sequence normalization process is as follows: the original cost sequence $\{C_t\}$ is numerically standardized to avoid gradient explosion and improve convergence performance. The transformation is defined as shown in Eq. (1) (Theodorou et al., 2025; Grooms et al., 2023).

$$C_t' = \frac{C_t - C_{\min}}{C_{\max} - C_{\min}} \tag{1}$$

In Eq. (1), C_t' is the normalized cost value, and $C_{\max}$ and $C_{\min}$ are the historical maximum and minimum cost values of the sequence, respectively. After normalization, the cost fluctuation pattern is preserved while ensuring the stability of the model training process. Accounting data often contains missing or outlier points due to business adjustments or accounting delays, which can easily interfere with cost trend learning (McClone et al., 2023; Barja-Martinez et al., 2024). Therefore, this study adopts a local statistical detection and smoothing correction strategy to suppress anomalies. The outlier identification conditions are shown in Eq. (2).

$$|C_t' - \mu| > 3\sigma \Rightarrow C_t' = \mu \tag{2}$$

In Eq. (2), μ and σ The mean and standard deviation of the sliding window are used, respectively, to assess the degree of cost deviation and to stabilize and replace extreme values. When constructing the model input, refer to the temporal convolution receptive field structure and divide the normalized sequence into continuous input segments using a sliding window, as shown in Eq. (3) (Bernknopf et al., 2025; Yin et al., 2024).

$$X_i = [C'_{1+(i-1)s}, \ldots, C'_{(i-1)s+W}], i = 1, 2, \ldots, T \tag{3}$$

In Eq. (3), X_i represents the i th input sample sequence, W is the window size, and T represents the number of samples generated by the sliding window. The calculation formula for T is shown in Eq. (4).

$$T = \frac{N - W}{s} + 1 \tag{4}$$

In Eq. (4), N is the sequence length, and s is the stride. Zero-padding ($P = 0$) is adopted to maintain historical causality. This mechanism preserves the local fluctuation structure, which helps temporal convolution extract short-term trend dependencies from cost changes. The corresponding future prediction label definition is shown in Eq. (5).

$$Y_i = [C'_{i+W}, \cdots, C'_{i+W+K-1}] \tag{5}$$

In Eq. (5), Y_i represents the prediction label. where K denotes the discrete prediction horizon. This multi-step labeling rule ensures the model captures continuous cost trajectories rather than a simple horizon average. The overall data preprocessing flow for intelligent cost prediction is shown in Fig. 1.

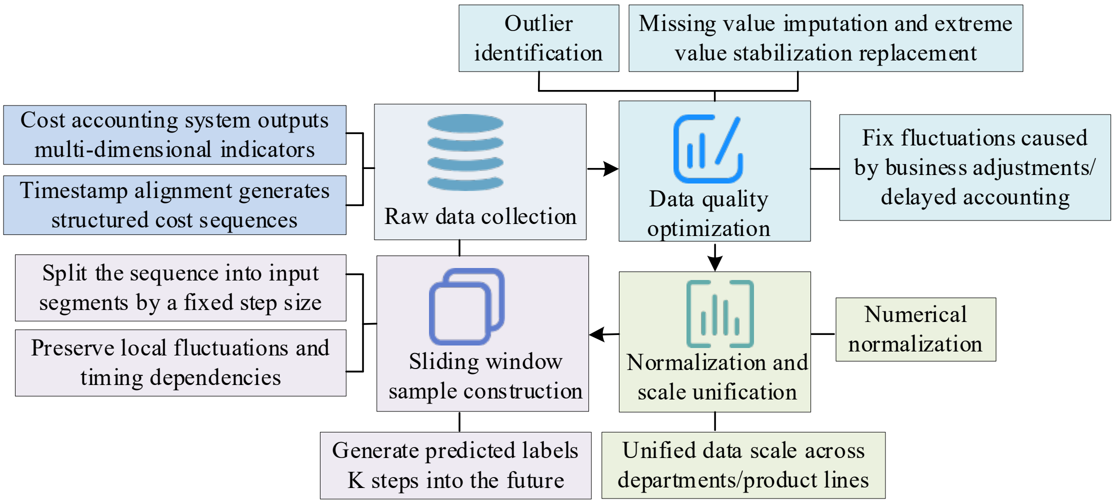


Fig. 1. Data preprocessing process for intelligent cost prediction

Fig. 1 shows the entire process for accounting for cost data, from raw data collection to model input. The process is divided into raw data collection, data quality optimization, normalization and scaling, and sliding window sample construction. Raw data collection includes multi-dimensional indicators, such as direct material costs, labor costs, and manufacturing expenses, generated by the cost accounting system, and forms a structured cost sequence through timestamp alignment (Oakley et al., 2024; Kuhlmann and Pauly, 2023). Data quality optimization uses anomaly identification and interpolation strategies to robustly repair data loss and abnormal fluctuations caused by business adjustments, delayed accounting, or calculation errors, making the sequence smoother and more predictable. In the normalization and scaling stage, cost data from different departments or product lines are standardized to avoid interference from scale differences during neural network training and to improve the model's generalization across business scenarios. The entire process suppresses random noise within a controllable range while preserving key financial characteristics. The study conducted a visual analysis of the temporal and distribution characteristics of the original cost data. The periodic trends and random disturbance characteristics of typical cost data are shown in Fig. 2.

Fig. 2 illustrates the patterns of variation in typical accounting cost data across different time spans. Cost values exhibit stable cyclical fluctuations from quarter to quarter or production cycle to production cycle, reflecting changes in seasonal operating strategies and resource allocation rhythms. After identifying the potential statistical structure of the cost series, it is necessary to further clarify the local dependency features at continuous-time nodes to ensure that the model can fully capture short-cycle fluctuation patterns. This study introduces a sliding window modeling strategy to segment long sequences into input segments with learnable semantics. Fig. 3 illustrates the specific construction method for the cost series sliding-window slices.

Fig. 3 illustrates the overall process of progressively sliding fixed-length windows over normalized cost time series data. Each window segment consists of continuous cost records from adjacent time steps, containing information on short-term trend changes, local peak-valley structures, and cost fluctuation amplitudes. As the window moves at a fixed step size, an overlapping set of input samples is formed, which not only effectively expands the amount of training data but also preserves the influence of historical information on future costs. This method simulates the thinking of corporate management when making budget and cost forecasts based on the cost performance of recent periods, enabling the model to learn decision-making basis from the cost status of neighboring countries (Koren and Shnaiderman, 2023; Zhang et al., 2023). After completing the window segmentation of the cost series, it is necessary to establish a unified data-processing pipeline to ensure global data quality consistency (Singh et al., 2025; Cao et al., 2023).

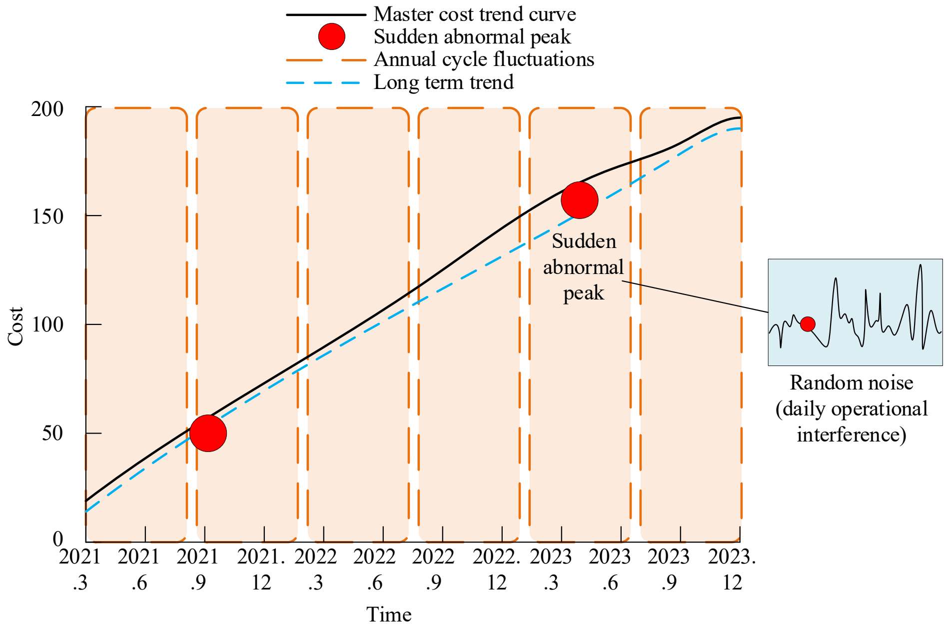


Fig. 2. Schematic diagram of statistical characteristics of accounting cost series

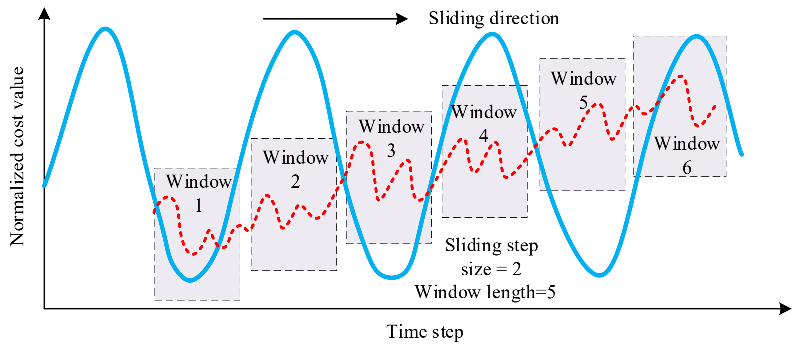


Fig. 3. Schematic diagram of cost sequence sliding window slicing

2.2. Construction of MTCN-CAM Intelligent Cost Accounting Prediction Model

By systematically designing a multi-dimensional statistical feature identification and data preprocessing process for accounting cost series, a stable and learnable data foundation was constructed. To further improve the accuracy and robustness of predictions, an intelligent model structure capable of simultaneously capturing multi-scale fluctuation features and key temporal dependencies was established, thereby achieving deep modeling and efficient prediction of complex accounting cost dynamics. To accurately capture the multi-scale dependency features and key fluctuation signals of accounting cost series, the MTCN-CAM model was proposed. Through multi-path feature extraction and dynamic weight allocation mechanisms, deep modeling of the trend, periodicity, and abrupt changes in cost data was achieved. The model as a whole sequentially maps input data to prediction results through multi-scale temporal convolutional blocks, a two-dimensional attention module, and a feature fusion layer. Its core structure is shown in Fig. 4.

Fig. 4 illustrates the hierarchical structure and data flow of the MTCN-CAM model. The model input is the sliding window cost sample sequence X_i (dimension $T \times L \times C$) after preprocessing in the previous section, and the output is the cost prediction value $\hat{Y}_i$ for the next K time steps. The data enters the multi-scale temporal convolution module, which extracts local dependency features at different time scales through parallel dilated convolution paths. The channel and temporal collaborative attention module dynamically enhances the weights of key features and suppresses redundant information interference. Finally, the result is produced by a feature fusion layer and a fully connected prediction head. This architecture preserves the fine-grained fluctuation details of the cost sequence while enhancing the modeling ability for long-term trends and key nodes. For the fluctuation patterns of accounting cost sequences across different time spans (such as weekly raw material procurement fluctuations and quarterly production cycle fluctuations), the MTCN module was designed to simultaneously extract multi-granular features via parallel dilated convolution paths, as shown in Fig. 5.

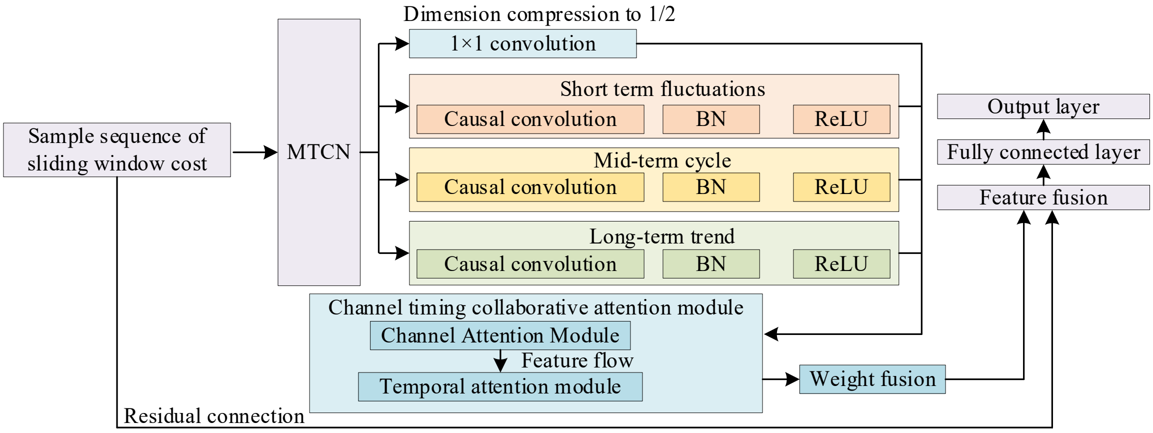


Fig. 4. Overall architecture diagram of MTCN-CAM model

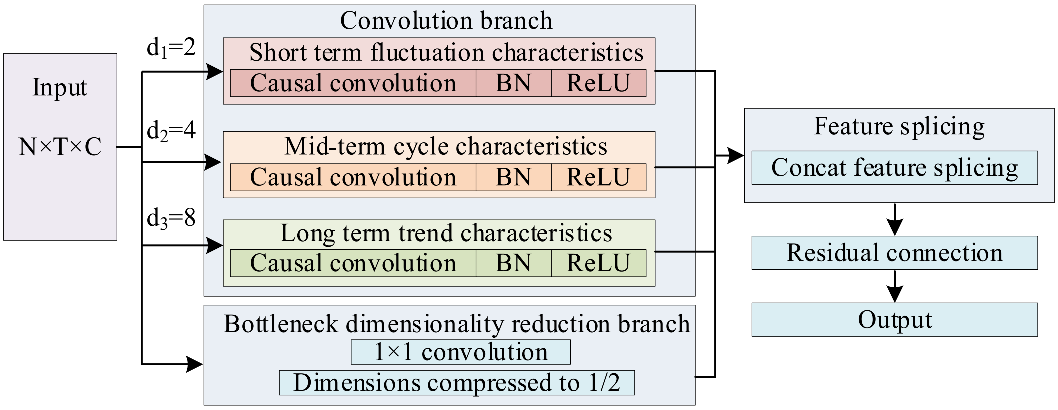


Fig. 5. MTCN feature extraction module structure diagram

In Fig. 5, the MTCN feature extraction module includes three parallel causal dilated convolution branches and one bottleneck dimensionality-reduction branch. Each convolution branch uses a different dilation rate d to construct a differentiated receptive field. Specifically, branch 1 (dilation rate $d_1 = 2$) focuses on short-term fluctuation features, with a receptive field covering 4 time steps; branch 2 (dilation rate $d_2 = 4$) captures medium-term cycle features, with a receptive field covering 8 time steps; and branch 3 (dilation rate $d_3 = 8$) extracts long-term trend features, with a receptive field covering 16 time steps. Each branch adopts a three-layer structure of causal convolution-BN-ReLU, where causal convolution ensures the directionality of time-series data and prevents future information leakage. Its calculation is shown in Eq. (6).

$$F_d(t) = \sum_{k=0}^{K'-1} g_k \cdot X_i(t - d \cdot (b)) \tag{6}$$

In Eq. (6), g_k represents the weight of the kth convolutional kernel, d is the dilation rate, and K' is the kernel size (takes a value of 3). $F_d(t)$ is the output feature of the convolutional branch with the dilation rate d at time t. The bottleneck dimensionality reduction branch compresses the input feature dimension to half of the original dimension through 1×1 convolution, reducing computational overhead. Its output is concatenated with the features of the three convolutional branches and returned to the module output through residual connection. The residual calculation is shown in Eq. (7).

$$F_{\text{mtcn}} = \sigma\left(W_{\text{r}} \cdot \text{Concat}(F_{d_1}, F_{d_2}, F_{d_3}, F_{\text{bn}}) + b_{\text{r}}\right) + (X) \tag{7}$$

In Eq. (7), Concat is the feature concatenation operation, F_{bn} is the output of the bottleneck dimensionality reduction branch, W_r is the residual mapping weight matrix, b_r is the residual bias term, and F_m is the final output feature of the module. This module can capture instantaneous fluctuations, periodic fluctuations and long-term trends in the cost sequence simultaneously by combining multi-scale receptive fields, addressing the limitation that single-scale convolution is insufficient for modeling complex financial data. Accounting cost data includes multiple indicator channels such as direct materials and labor costs, and the influence weights of different time nodes on the prediction results are significantly different (Tian and Liu, 2024; Xu et al., 2023). To this end, a channel-time series collaborative attention module was designed to dynamically allocate feature weights from two dimensions: indicator importance and time correlation. Its structure is shown in Fig. 6.

Fig. 6 shows that the module is composed of a channel attention submodule and a temporal attention submodule connected in series. The channel attention submodule evaluates the predictive contribution of each cost indicator through a squeeze-incentive mechanism. First, it performs global average pooling on the input feature F_m to compress the temporal dimension to 1, as shown in Eq. (8).

$$S_c = \frac{1}{W}\sum_{t=1}^{W} F_{\mathrm{m}}(i,t,c)(i=1,2,...,T) \tag{8}$$

In Eq. (8), S_c is the global statistical value of the c th channel. A channel weight learning network is constructed through two fully connected layers to output the normalized weights of each channel, as shown in Eq. (9).

$$\alpha_c = \delta\left(W_2 \cdot \sigma\left(W_1 \cdot S_c + b_1\right) + b_2\right) \tag{9}$$

In Eq. (9), W_1 is the weight matrix of the first-level fully connected layer, and W_2 is the weight matrix of the second-level fully connected layer. b_1 and b_2 are the bias terms of the two fully connected layers, respectively, and α_c is the weight value of the c th channel. The temporal attention submodule uses the dot product attention mechanism to calculate the association weights at different time steps. First, the input features are converted into query vector $\mathbf{Q}$, key vector $\mathbf{K}$, and value vector $\mathbf{V}$, and then the temporal weights are obtained through similarity calculation and normalization, as shown in Eq. (10).

$$\beta_t = \mathrm{Softmax}\left(\frac{Q \cdot K^T}{\sqrt{C_k}}\right)\mathbf{V} \tag{10}$$

In Eq. (10), $\mathbf{K}^T$ is the transpose of the key vector, $\sqrt{C_k}$ is the scaling factor, and β_t is the weight value at the t th time step. Finally, feature enhancement is achieved through the element-wise product of channel weights and temporal weights. To integrate the complementary information of multi-scale convolutional features and attention-enhanced features, a residual feature fusion layer and a multi-step prediction head are designed to realize the mapping from high-dimensional features to cost prediction values. Its structure is shown in Fig. 7.

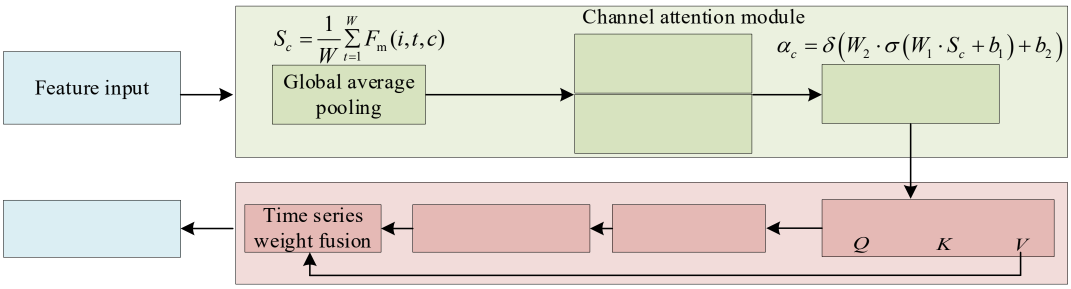


Fig. 6. Channel-sequential collaborative attention module structure diagram

Fig. 7 illustrates the structure of the feature fusion layer and the prediction output layer. This layer concatenates the multi-scale temporal convolution output and the collaborative attention output along the channel dimension to obtain fused features. The feature dimension is reduced to half of the original dimension through 1×1 convolution, reducing the parameter size. Finally, a residual connection is established with the input features to alleviate the gradient vanishing problem in deep networks. The prediction output layer adopts a stacked fully connected structure, which maps the fused features to the cost prediction values for future K time steps through two-level linear transformations. At the same time, L2 regularization is introduced to suppress overfitting. The prediction calculation is shown in Eq. (11).

$$\hat{Y}_i = W_{p2} \cdot \sigma\left(W_{p1} \cdot F' + b_{p1}\right) + b_{p2} + \lambda \cdot \sum \|W\|^2 \tag{11}$$

In Eq. (11), F' represents the flattened result of the fused features. W_{p1} is the weight matrix of the first-level fully connected layer, and W_{p2} is the weight matrix of the second-level fully connected layer. b_{p1} and b_{p2} are the bias terms of the two fully connected layers, respectively. λ is the L2 regularization coefficient (with a value of 10^{-4}). $\hat{Y}_i$ is the final predicted sequence for the i th sample. This output structure can flexibly adapt to different financial forecasting scenarios such as short-term (K=7, weekly forecast), medium-term (K=30, monthly forecast), and long-term (K=90, quarterly forecast), meeting the needs of multi-dimensional cost control.

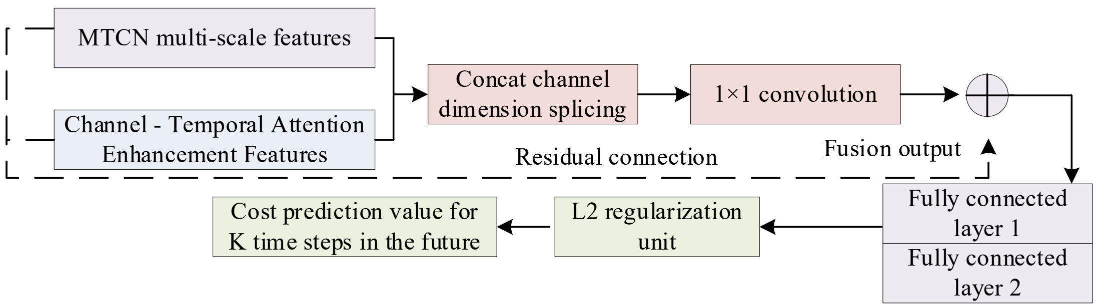


Fig. 7. Feature fusion and prediction output layer structure

3. Results

3.1. Experimental Setup and Model Performance Analysis

To verify the performance advantages of the proposed MTCN-CAM model in intelligent prediction of accounting costs, a comparative experiment was designed, and training and testing were completed in a unified environment to ensure the objectivity and reproducibility of the results. The experimental hardware platform was equipped with an NVIDIA RTX 4090 (24GB VRAM) GPU, an Intel Core i9-13900K processor, and 128GB of RAM. The software environment included an Ubuntu 22.04 operating system, the PyTorch 2.2 deep learning framework, CUDA 12.1, and the cuDNN 9.0 acceleration library. A mixed-precision training and gradient pruning strategy was enabled to improve the model's convergence stability. The dataset came from real business data in the cost accounting system of a manufacturing group, involving 3 subsidiaries, 12 cost centers, and 72 monthly cost indicators from 2019 to 2024, including 12 dimensions such as direct materials, labor costs, and manufacturing overhead. The pooled approach aggregates samples from all 3 subsidiaries into a unified temporal stream, ensuring the model learns universal cost-driver relationships while maintaining strict temporal separation to prevent data leakage. The dataset was partitioned using a pooled chronological split to capture cross-entity cost patterns: the training set covers Jan 2019–Dec 2022, the validation set covers Jan 2023–Dec 2023, and the test set covers the rolling periods of 2024 (Q1–Q4). The comparison models selected were Autoregressive Integrated Moving Average (ARIMA), 1D Convolutional Neural Network (1D-CNN), and Temporal Fusion Transformer (TFT), comparing three paradigms from the perspectives of statistical models, convolutional structures, and attention mechanisms. The comparison results of the prediction accuracy of different models are shown in Fig. 8.

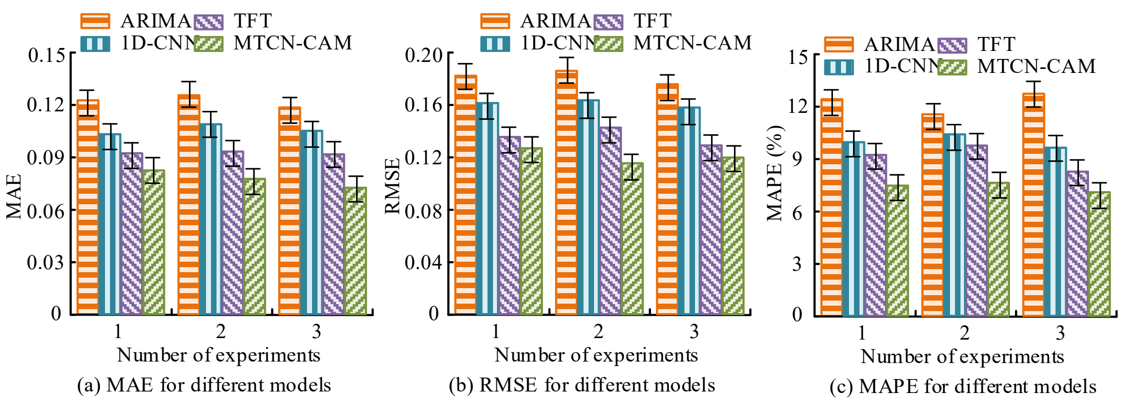


Fig. 8. Comparison of prediction accuracy of different models

As shown in Fig. 8(a), the proposed MTCN-CAM model performed best in terms of Mean Absolute Error (MAE). Taking the results of the third experiment as an example, the MAE of MTCN-CAM was 0.07, which was significantly lower than that of ARIMA (0.11), 1D-CNN (0.09), and TFT (0.08). In the three sets of experiments, the average reduction in error of MTCN-CAM was 34.1% relative to ARIMA, 18.2% relative to 1D-CNN, and 10.9% relative to TFT, respectively, indicating that the model could more accurately fit the real change structure of accounting costs, thereby effectively suppressing the accumulation of systematic errors caused by noise interference. As shown in Fig. 8(b), MTCN-CAM maintained the best and most stable performance in terms of Root Mean Square Error (RMSE). The results of the third experiment showed that the RMSE of MTCN-CAM was 0.12, which was better than that of ARIMA (0.17), 1D-CNN (0.14), and TFT (0.13). As shown in Fig. 8 (c), MTCN-CAM also exhibited a significant advantage in Mean Absolute Percentage Error (MAPE), with a minimum MAPE of 8.50%, outperforming ARIMA (12.10%), 1D-CNN (10.20%), and TFT (9.10%). These results showed that the MTCN-CAM demonstrated a significant competitive advantage in recognition accuracy and is suitable for deployment in actual financial cost prediction and budget management operations. The robustness evaluation results of different models are shown in Fig. 9.

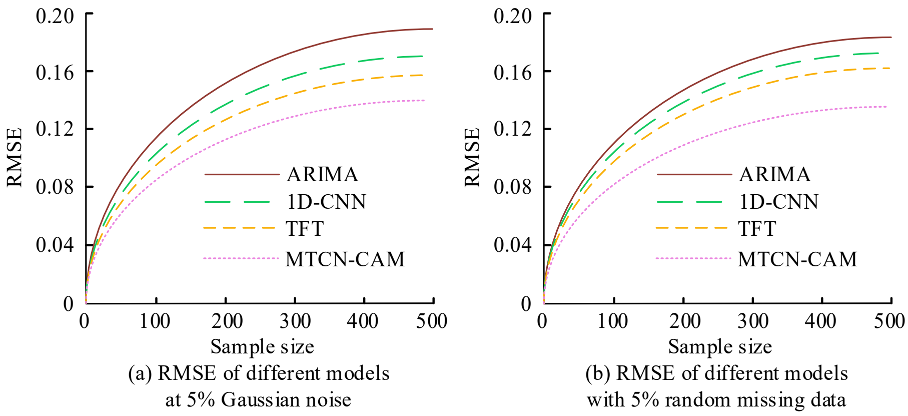


Fig. 9. Robustness evaluation results of different models

As shown in Fig. 9(a), the MTCN-CAM error curve showed a significant downward trend as the sample size increased, compared with ARIMA, 1D-CNN, and TFT. When the sample size reached 500, the RMSE of MTCN-CAM stabilized below 0.08, while the RMSEs of TFT, 1D-CNN, and ARIMA were 0.12, 0.13, and 0.14, respectively. As shown in Fig. 9(b), when the sample size reached 500, the RMSE of MTCN-CAM was 0.09, significantly lower than those of the other baseline models. ARIMA's RMSE was close to 0.16, while 1D-CNN and TFT were 0.14 and 0.13, respectively. MTCN-CAM's error reduction was 37.5% compared to these models. This result further verified that MTCN-CAM had strong robustness and excellent feature recovery ability when dealing with missing data, especially in real-world applications where missing data was a common phenomenon MTCN-CAM's performance was more stable. Experimental data showed that the MTCN-CAM model could maintain high prediction accuracy and reliability even in unstable environments. This was mainly due to MTCN-CAM's multi-scale convolutional feature extraction mechanism and channel-temporal collaborative attention module, which enabled the model to dynamically adjust feature importance and effectively preserve core information in the presence of noise or missing data. The comparison of the prediction efficiency of different models is shown in Fig. 10.

As shown in Fig. 10(a), ARIMA's throughput reached a peak of 1200 samples/second with a sample size of 200 and remained around 1100 samples/second, indicating that the model was optimal for processing small-scale data. In contrast, the throughput of 1D-CNN and TFT was significantly lower and decreased with the increase of the number of samples, especially TFT, whose throughput decreased significantly with the increase of samples, from 900 samples/second to 500 samples/second, indicating that its computational requirements were higher and its processing capacity was affected. The throughput of MTCN-CAM remained relatively stable at about 600 samples/second, indicating higher throughput efficiency than 1D-CNN and TFT. As shown in Fig. 10(b), the GPU utilization of TFT and MTCN-CAM gradually increased with the increase in the number of samples. ARIMA's GPU utilization remained low, peaking at 20%, indicating the model was suitable for low-resource scenarios. In Fig. 10(c), ARIMA's inference latency remained below 1ms as the number of samples increased. 1D-CNN had a relatively high inference latency, especially with a large sample size, increasing from 2.2ms to 3.5ms. TFT's inference latency changed more significantly, increasing from 3.8ms to 5ms. Ablation experiments are shown in Table 1.

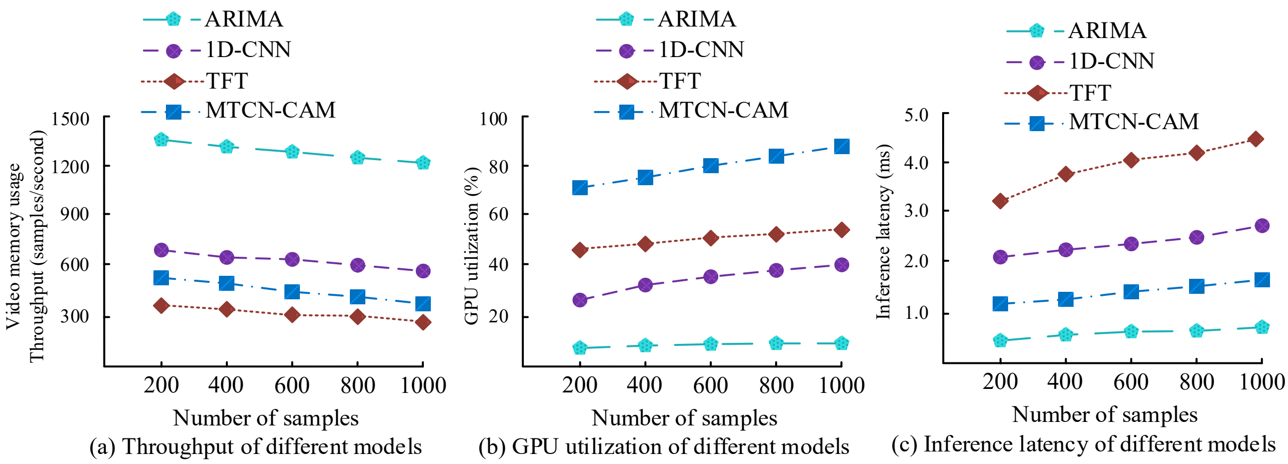
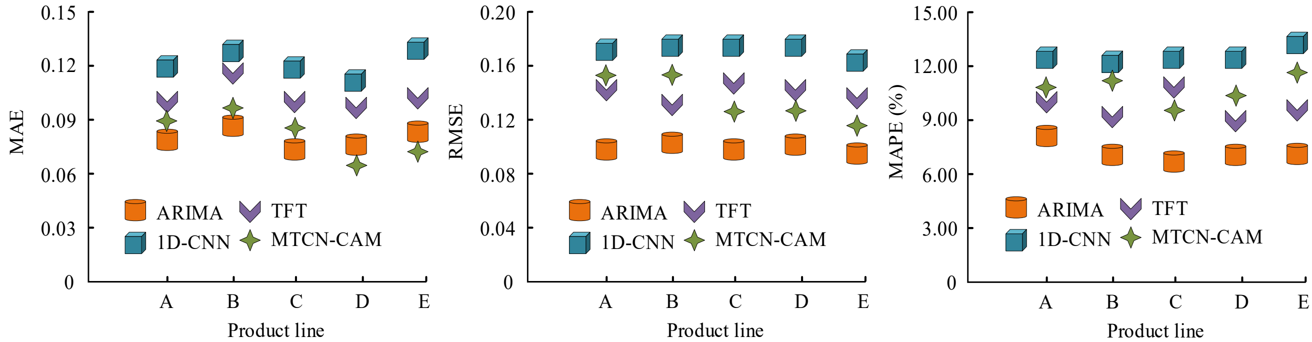


Fig. 10. Comparison results of prediction efficiency of different models

Table 1. Ablation experiment

Model	MAE	RMSE	MAPE (%)	FLOPs / Samples (G)	Inference latency (ms)
MTCN-CAM	0.07	0.12	8.32	0.06	2.15
Remove multi-scale convolution module	0.09	0.14	9.75	0.08	2.47
Remove attention module	0.08	0.13	8.92	0.07	2.32
Remove feature fusion module	0.08	0.13	9.18	0.07	2.38
Remove convolution and attention modules	0.11	0.15	11.24	0.09	2.56
Remove multi-scale convolution, attention and fusion modules	0.12	0.16	12.65	0.10	2.68

In Table 1, the complete MTCN-CAM model had an MAE of 0.07, RMSE of 0.12, and MAPE of 8.32%. Removing the multi-scale convolution module decreased model accuracy, while MAE increased to 0.09, RMSE to 0.14, and MAPE to 9.75%. Removing the attention module maintained a relatively stable model performance, but accuracy decreased slightly. In particular, removing the convolution and attention modules improved MAE to 0.11, RMSE to 0.15, and MAPE to 11.24%. Regarding inference latency, it increased with module removal, reaching 2.68ms when all modules were removed. Overall, the modules of MTCN-CAM played a crucial role in maintaining model accuracy and robustness.

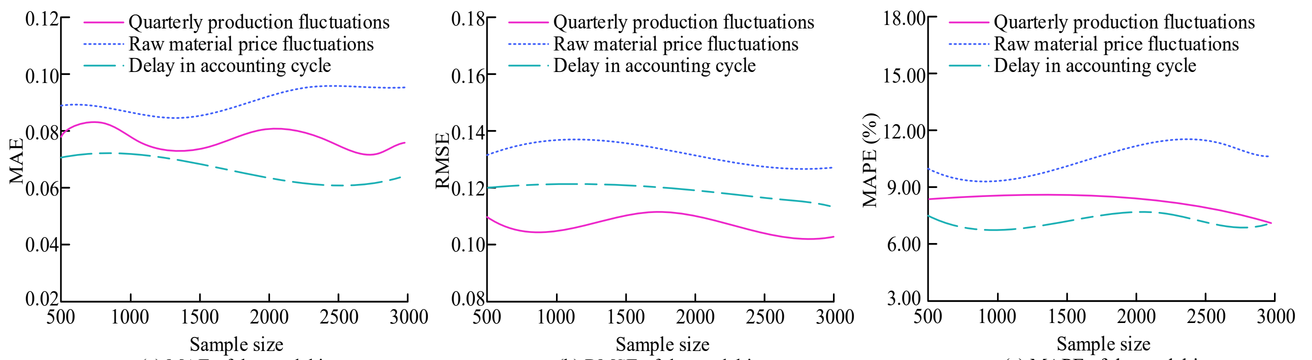


(a) MAE of different models in cost prediction of different product lines
(b) RMSE of different models in cost prediction of different product lines
(c) MAPE of different models in cost prediction of different product lines

Fig. 11. Comparison of the accuracy of different models in multi-product line cost prediction

3.2. Accounting Cost Forecast Results and Comparative Analysis

Fig. 11 shows the comparison of the accuracy of different models in multi-product line cost forecasting. In Fig. 11(a), the MTCN-CAM model achieved the lowest MAE across all five product lines (A-E). Fig. 11(b) shows that MTCN-CAM also exhibited the most stable performance in terms of RMSE, with its RMSE consistently below 0.12. Fig. 11(c) shows that MTCN-CAM maintained the best performance in terms of MAPE, with errors concentrated in the 8%-9% range across product lines. Experimental data showed that MTCN-CAM demonstrated higher prediction accuracy, smaller error fluctuations, and stronger business adaptability. The stability test results of the MTCN-CAM model in complex business environments are shown in Fig. 12.



(a) MAE of the model in different business scenarios
(b) RMSE of the model in different business scenarios
(c) MAPE of the model in different business scenarios

Fig. 12. Stability of different models in different business scenarios

In Fig. 12(a), the MTCN-CAM model maintained an error of around 0.08 in both the "quarterly output fluctuation" and "receipt delay" scenarios. In Fig. 12(b), the overall error fluctuation range of MTCN-CAM was controlled between 0.10 and 0.13. In Fig. 12(c), in the "receipt delay" scenario, the MAPE of MTCN-CAM remained consistently below 8%. In the "raw material price fluctuations" scenario, although the MAPE was slightly higher, it still consistently outperformed other models.

MTCN-CAM exhibited excellent stability, error-control capabilities, and model generalization performance across different business-risk conditions. It had significant application value for actual budget management and cost risk early warning. Table 2 compares the MAPE of different models for predicting cost components.

Table 2. Comparison of MAPE for cost component prediction by different models

Model	Direct materials (%)	Direct labor (%)	Manufacturing overhead (%)	Energy consumption (%)	Logistics and transportation (%)	Equipment depreciation (%)	Administrative expenses (%)	Average MAPE (%)
MTCN-CAM	8.10	8.60	8.90	9.20	8.40	8.00	8.70	8.56
ARIMA	12.60	12.10	13.40	14.20	12.80	13.00	12.50	12.94
1D-CNN	0.80	11.20	10.60	11.50	10.40	10.90	11.10	10.93
TFT	10.20	10.50	11.10	11.80	9.90	10.30	10.70	10.64

Table 2 shows that the MTCN-CAM model achieved the lowest MAPE across all cost components, with an average of 8.56%. For factors that significantly impact manufacturing costs, such as equipment depreciation and direct materials, MTCN-CAM achieved values of 8.00% and 8.10%, respectively. Furthermore, the ARIMA model's MAPE exceeded 12% across all factors, with an energy consumption error of 14.20%. The experimental results demonstrated that MTCN-CAM's ability to identify cost fluctuations and its business adaptability under multiple cost factors were significantly superior to those of the comparative models. Table 3 compares the prediction biases of different models across different prediction periods.

Table 3. Comparison of prediction deviations of different models under different prediction periods

Model	Ultra-short (Daily)	Short-term (Weekly)	Mid-term (Monthly)	Long-term (Quarterly)	Very-long (Semiannual)	MAE (%)
MTCN-CAM	-0.21	-0.35	0.28	-0.42	0.15	0.28
ARIMA	1.35	1.48	2.12	1.73	1.95	1.73
1D-CNN	-0.76	-0.92	0.97	0.41	-0.15	0.64
TFT	0.52	0.63	1.24	0.79	0.88	0.81

Note: Positive = overestimation, negative = underestimation.

Table 3 shows that MTCN-CAM had the smallest prediction bias among the four models: -0.35% for weekly, 0.28% for monthly, 0.42% for quarterly, and 0.15% for semi-annual, with an MAE of 0.30%. Experimental data showed that MTCN-CAM exhibited low bias, weak directionality, and high stability across short, medium, long, and ultra-long periods. To further demonstrate the practical impact beyond statistical errors, this study evaluated the models under a "Material Price Shock" managerial scenario. In this scenario, predictive performance was mapped to decision-relevant metrics: Cost Efficiency (capital buffer optimization), Timeliness (inference latency), and Decision Reliability (error stability under 10% noise). The results are summarized in Table 4.

Table 4. Comparison of decision-relevant metrics under "Material Price Shock" scenario

Model	MAE	Cost Efficiency (Capital Saving %)	Decision Latency (ms)	Reliability (RMSE in Noise)
MTCN-CAM	0.07	4.2	2.2	0.08
TFT	0.08	3.5	4.1	0.13
XGBoost	0.10	2.8	1.5	0.15
LSTM	0.11	2.4	3.2	0.14
Prophet	0.13	1.8	12.5	0.18
Naïve Seasonal	0.16	0.5	0.1	0.22

As shown in Table 4, MTCN-CAM translated its predictive superiority into a 4.2% capital saving by minimizing budget slack, a critical advantage for lean management. Unlike complex attention models, TFT or slow statistical tools (Prophet), MTCN-CAM maintained sub-3ms latency and the highest reliability in noisy scenarios, providing a stable foundation for real-time procurement and safety-stock adjustments.

4. Discussion

Compared with existing studies, the model by Damavandi et al. (2025), based on EVM and Markov chains, was limited in its ability to handle sudden exogenous shocks and seasonal fluctuations. Wang and Zhao's (2023) convolutional Transformer framework captured long- and short-term dependencies but lacked explicit modeling for multi-indicator cost channels and incurred high inference overhead. In contrast, MTCN-CAM achieved unified modeling of multi-scale fluctuations via parallel dilated convolution and enhanced the representation of key cost elements using collaborative attention. Experiments demonstrated that MTCN-CAM maintained an RMSE of 0.08-0.09 even in noisy and data-missing scenarios, exhibiting superior robustness and real-time performance, which better aligned with the needs of enterprise rolling budgets and risk monitoring. The MTCN-CAM model acted as a bridge between operational data and strategic control. By providing a low-bias "future trajectory," it empowered CFOs to refine rolling budgets in real-time, thereby optimizing the trade-off between inventory costs and production stability. However, it is worth noting that for specific cost centers with low volatility and stable linear trends (e.g., fixed equipment depreciation), simpler models like ARIMA or Naive Seasonal may suffice due to their lower computational footprint and higher interpretability. The complexity of MTCN-CAM is most justified in environments characterized by high-frequency disturbances and multi-factor coupling. Nevertheless, the research still lacked sufficient cross-industry generalizability. In the future, the study will combine self-supervised learning and causal inference to expand the model's adaptability and explore edge deployment and exogenous indicator fusion strategies to further improve forward prediction performance in resource-constrained environments and cost-shock scenarios.

5. Conclusion

Traditional models struggle to capture both short-term fluctuations and medium-to-long-term trends in accounting cost series, which exhibit complex characteristics, including trend, seasonality, suddenness, and asynchronous changes. To address this, the MTCN-CAM model was developed. This model uses parallel dilated convolution to uniformly model change patterns across different business cycles and employs dual-dimensional attention to enhance the representation of key cost elements and critical time points. Experimental results showed that MTCN-CAM outperformed ARIMA, 1D-CNN, and TFT in overall prediction accuracy. The MAE across the four forecasting periods was 0.30%, effectively avoiding systematic overestimation or underestimation. The inference latency was approximately 2.0ms to 2.8ms, meeting the high-efficiency forecasting requirements for real-time cost control and risk warning in enterprises. Overall, MTCN-CAM balanced accuracy, robustness, and low latency for lean accounting management. MTCN-CAM facilitated a fundamental shift in managerial decision processes. Specifically, it enabled managers to transition from "ex-post" reactive adjustments to "ex-ante" proactive steering by providing millisecond-level updates for rolling budgets. Managers can now replace static annual allocations with dynamic resource rebalancing, where procurement and production schedules are adjusted based on predicted cost trajectories rather than historical averages. This data-driven approach minimizes budget slack and ensures that cost-risk interventions are triggered by structural trends rather than accounting noise, directly enhancing the agility and lean performance of enterprise governance. However, its performance may degrade under significant domain shifts or black-swan exogenous shocks, as it currently relies on historical internal data. Future work will focus on integrating self-supervised pretraining and causal feature extraction to enhance generalization. Additionally, incorporating external indices, such as raw material commodity prices and macroeconomic indicators, will be explored to improve the model's proactive response to global market volatility.

Author Contributions

Tong Wu contributes to conceptualization, methodology, software, validation, and draft preparation. Shuang Wu contributes to investigation, data collection, supervision and manuscript editing. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

Declaration of Artificial Intelligence (AI) Tools

The authors used Doubao solely for language editing and to improve readability. The authors reviewed and verified all content and take full responsibility for the accuracy and integrity of the manuscript.

References

Ahabchane, C., Cenesizoglu, T., Grass, G., and Jena, S. D. (2024). Reducing transaction costs using intraday forecasts of limit order book slopes. *Journal of Forecasting,* 43(8), 2982–3008. doi: 10.1002/for.3164

Aydinli, S. (2024). Impact of unexpected conditions on construction cost forecasting performance: evidence from Europe. *Construction Management and Economics*, 42(9), 787–801. doi: 10.1080/01446193.2024.2328700

Barja-Martinez, S., Teng, F., Junyent-Ferré, A., and Aragüés-Peñalba, M. (2024). Personalized federated learning with cost-oriented load forecasting for home energy management systems. *IEEE Transactions on Industry Applications,* 61(1), 1410–1419. doi: 10.1109/TIA.2024.3462668

Bernknopf, R., Kuwayama, Y., Zaitchik, B., Rodell, M., Getirana, A., Thorstensen, A., and Shahreen, S. (2025). An economic impact assessment of the use of earth observation information in flood hazard communication. *Natural Hazards,* 121(15), 17913–17933. doi: 10.1007/s11069-025-07499-3

Boone, J. P., Sun, X., and Yin, J. (2024). Information or noise? Examining the effect of discretionary disclosure of accounting estimates on financial analyst forecasts. *Abacus*, 60(4), 852–891. doi: 10.1111/abac.12343

Cao, H., Wu, Y., Bao, Y., Feng, X., Wan, S., and Qian, C. (2023). UTrans-Net: A model for short-term precipitation prediction. *Artificial Intelligence and Applications,* 1(2), 106–113. doi: 10.47852/bonviewAIA2202337

Chen, Y., Sun, M., Zchu, Z., Camal, S., Kariniotakis, G., and Teng, F. (2022). Vulnerability and impact of machine learning-based inertia forecasting under cost-oriented data integrity attack. *IEEE Transactions on Smart Grid,* 14(3), 2275–2287. doi: 10.1109/TSG.2022.3207517

Damavandi, M., Tavakoli, M., and Jolai, F. (2025). Project cost forecasting based on earned value management and Markov chain. *Annals of Operations Research*, 351(1), 1023–1048. doi: 10.1007/s10479-024-05889-7

Dossatayev, A., Manapova, A., and Omarov, B. (2023). Deep residual convolutional long short-term memory network for option price prediction problem. *International Journal of Advanced Computer Science and Applications,* 14(9), 377–387. doi: 10.14569/IJACSA.2023.0140941

Easton, P. D., Kapons, M. M., Monahan, S. J., Schütt, H. H., and Weisbrod, E. H. (2024). Forecasting earnings using k-nearest neighbors. The Accounting Review, 99(3), 115–140. doi: 10.2308/TAR-2021-0478

Ghadimi, S. and Powell, W. B. (2024). Stochastic search for a parametric cost function approximation: Energy storage with rolling forecasts. *European Journal of Operational Research,* 312(2), 641–652. doi: 10.1016/j.ejor.2023.08.003

Grooms, I., Renaud, C., Stanley, Z., and Yang, L. M. (2023). Analog ensemble data assimilation in a quasigeostrophic coupled model. *Quarterly Journal of the Royal Meteorological Society,* 149(752), 1018–1037. doi: 10.1002/qj.4446

Hu, Z., Yang, R., and Borjigin, S. (2024). A multistage forecasting model for green bond cost optimization with dynamic corporate risk constraints. *Journal of Forecasting*, 43(7), 2607–2634. doi: 10.1002/for.3142

Huang, P. (2023). Internet financial forecasting and digital economy development by using machine learning algorithm in the new consumption environment. *Soft Computing,* 27(14), 10285–10296. doi: 10.1007/s00500-023-08309-3

Koren, M. and Shnaiderman, M. (2023). Forecasting in the fashion industry: a model for minimising supply-chain costs. *International Journal of Fashion Design, Technology and Education,* 16(3), 308–318. doi: 10.1080/17543266.2023.2201508

Kuhlmann, L. and Pauly, M. (2023). A dynamic systems model for an economic evaluation of sales forecasting methods. *Tehnički glasnik*, 17(3), 397–404. doi: 10.31803/tg-20230511175500

McClone, G., Ghosh, A., Khurram, A., Washom, B., and Kleissl, J. (2023). Hybrid machine learning forecasting for online MPC of workplace electric vehicle charging. *IEEE Transactions on Smart Grid,* 15(2), 1891–1901. doi: 10.1109/TSG.2023.3296014

Narbaev, T., Hazir, Ö., Khamitova, B., and Talgat, S. (2024). A machine learning study to improve the reliability of project cost estimates. *International Journal of Production Research,* 62(12), 4372–4388. doi: 10.1080/00207543.2023.2262051

Oakley, J., Conlan, C., Demirci, G. V., and Sfyridis, A. (2024). Foresight plus: serverless spatio-temporal traffic forecasting. *GeoInformatica,* 28(4), 649–677. doi: 10.1007/s10707-024-00517-9

Olson, G. T. and Pagano, M. S. (2024). Applying the empirical average cost of capital: Estimating the cost of funds at the firm and industry levels. *Applied Economics,* 56(49), 5921–5938. doi: 10.1080/00036846.2023.2266603

Patel, M., Jariwala, K., and Chattopadhyay, C. (2024). An approach toward stock market prediction and portfolio optimization in Indian financial sectors. *IEEE Transactions on Computational Social Systems,* 12(1), 128–139. doi: 10.1109/TCSS.2024.3450291

Petropoulos, F., Grushka-Cockayne, Y., Siemsen, E., and Spiliotis, E. (2025). Wielding Occam's razor: Fast and frugal retail forecasting. *Journal of the Operational Research Society*, 76(8), 1564–1583. doi: 10.1080/01605682.2024.2421339

Singh, C., Sreekumar, S., and Malakar, T. (2025). A novel dynamic imbalance volume forecasting model for balancing market optimization. *Electrical Engineering,* 107(12), 15375–15392. doi: 10.1007/s00202-025-03331-0

Theodorou, E., Spiliotis, E., and Assimakopoulos, V. (2025). Forecast accuracy and inventory performance: Insights on their relationship from the M5 competition data. *European Journal of Operational Research,* 322(2), 414–426. doi: 10.1016/j.ejor.2024.12.033

Tian, Y. and Liu, G. (2024). Spatial-temporal-aware graph transformer for transaction fraud detection. *IEEE Transactions on Industrial Informatics*, 20(11), 12659–12668. doi: 10.1109/TII.2024.3423447

Ünsal-Altuncan, I. and Vanhoucke, M. (2024). A hybrid forecasting model to predict the duration and cost performance of projects with Bayesian Networks. *European Journal of Operational Research,* 315(2), 511–527. doi: 10.1016/j.ejor.2023.12.029

Wallis, M. (2023). Why do analysts use a zero forecast for other comprehensive income? *Abacus,* 59(4), 1074–1115. doi: 10.1111/abac.12292

Wang, J., Qu, Z., Lee, C. Y., and Skitmore, M. (2025). Highway construction cost index forecasting: a hybrid VMD–LSTM–GRU method. *Construction Management and Economics,* 43(10), 849–863. doi: 10.1080/01446193.2025.2525871

Wang, N. and Zhao, X. (2023). Time series forecasting based on convolution transformer. *IEICE Transactions on Information and Systems,* 106(5), 976–985. doi: 10.1587/transinf.2022EDP7136

Xu, Z., Yu, Z., Zhang, H., Chen, J., Gu, J., and Lukasiewicz, T. (2023). PhaCIA-TCNs: Short-term load forecasting using temporal convolutional networks with parallel hybrid activated convolution and input attention. *IEEE Transactions on Network Science and Engineering,* 11(1), 427–438. doi: 10.1109/TNSE.2023.3300744

Yan, X., Zhang, H., Li, R., Wan, C., Tai, N., and Sun, Y. (2024). A cost-oriented score evaluation index to align probabilistic forecasting and uncertain decision-making in power systems. *IEEE Transactions on Power Systems,* 40(2), 1981–1984. doi: 10.1109/TPWRS.2024.3522708

Yin, W., Chen, Z., Luo, X., and Kirkulak-Uludag, B. (2024). Forecasting cryptocurrencies' price with the financial stress index: a graph neural network prediction strategy. *Applied Economics Letters*, 31(7), 630–639. doi: 10.1080/13504851.2022.2141436

Zhang, Y., Wen, H., and Wu, Q. (2023). A contextual bandit approach for value-oriented prediction interval forecasting. *IEEE Transactions on Smart Grid*, 15(2), 2271–2281. doi: 10.1109/TSG.2023.3296577

Tong Wu was born on August 14, 2006, in Qiqihar City, Heilongjiang Province, China. He has been studying at the Harbin University of Commerce since 2024, majoring in Accounting. He has published two academic papers, two research projects and received two academic awards.

Shuang Wu was born on May 10, 2003, in Qiqihar City, Heilongjiang Province, China. She has been studying at Beijing Wuzi University since 2020, majoring in Marketing.