

Article

Dynamic Cost Prediction for State Grid Engineering Projects Based on Multi-Source Business Data Fusion and Data-Driven Methods

Wei Qiong Wang ¹, Qidong Xu ¹, Tianyu Zhao ^{2,*} and Fang Fang ²

¹ Finance and Assets Department, State Grid Jiangxi Electric Power Co., Ltd., Nanchang 330077, China; weiqiong.wang2026@163.com (W.W.); 18707039409@163.com (Q.X.)

² Accounting Development Research Center, Jiangxi University of Finance and Economics, Nanchang 330013, China; fangfang@jxufe.edu.cn

* Correspondence: zhaotianyu@jxufe.edu.cn

Abstract

Accurate dynamic cost prediction is essential for budget optimization and risk mitigation in State Grid projects. However, traditional models and even recent deep learning approaches fall short, as they treat cost drivers independently, adopt simplistic concatenation that destroys sourcewise structure, or fail to handle irregularly sampled and partially missing multi-source data. This paper proposes a novel data-driven framework that integrates multi-source business data through a hierarchical tensor fusion mechanism and a hybrid spatiotemporal architecture. The problem is formalized as multivariate time-series prediction with irregular sampling and missing modalities. The framework comprises three synergistic innovations: a differentiable low-rank CANDECOMP/PARAFAC (CP) decomposition layer with adaptive attention weights that preserves cross-source structure while enabling compact dimensionality reduction; a spatiotemporal attention-based bidirectional gated recurrent unit (Bi-GRU) that captures long-range temporal dependencies; and a graph convolutional network (GCN) that explicitly learns interrelations among cost drivers, a capability absent in most existing forecasting methods. The entire system is trained end to end with a customized loss combining mean squared error, quantile loss, and temporal consistency regularization. Extensive experiments on three State Grid substation projects demonstrate that the proposed method outperforms state-of-the-art baselines by 12.7–18.4% in MAPE and maintains robust performance with up to 40% of data missing. These results confirm that explicitly modeling both temporal evolution and driver interdependencies within a unified fusion framework is the key to reliable cost forecasting in large-scale infrastructure projects.

Keywords: dynamic cost prediction; multi-source data fusion; graph convolutional network; bidirectional gated recurrent unit; spatiotemporal attention



Academic Editor: Haridimos Kondylakis

Received: 17 June 2026

Revised: 5 July 2026

Accepted: 13 July 2026

Published: 16 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

State Grid engineering projects (high-voltage substations and transmission lines) feature large capital outlays, extended cycles, and strict quality/safety requirements [1–3]. Life-cycle cost estimation and dynamic prediction are essential for budgeting, risk management, and resource allocation. However, costs are influenced by multiple time-varying factors—material prices, labor, equipment, weather, progress metrics, and

policy changes—which come from heterogeneous systems, have mismatched sampling frequencies (hourly to monthly), and suffer from missing data. This leads to highly non-linear cost dynamics with strong interdependencies among drivers.

Traditional cost estimation methods can be categorized into three classes. The first class comprises deterministic approaches such as the unit price method [4], parametric estimation [5], and analogy-based models [6]. In ref. [7], Esi et al., addressing the production pricing and cost of sale challenges for D’atabajo coffee, proposed using the full costing method to calculate production prices and the cost plus pricing method to determine the cost of sale. In ref. [8], Hu et al. proposed an adaptive deep load predictor (ADLP) that integrated mixed time series coding, dynamic attention mechanism and online adaptive updating, and they designed a dynamic grid perceived load optimization strategy (DGLO). Through prediction-driven control and robust optimization, accurate power load prediction and efficient optimization of power grid operation were realized. Experiments verified its superiority in accuracy, adaptability and robustness. These methods rely on fixed cost-driver relationships derived from historical standards or expert rules; thus, they fail to adapt to dynamic on-site variations and cannot capture non-linear interactions [9–11]. The second class includes statistical time-series models such as ARIMA [12], exponential smoothing [13], and regression analysis [14]. In ref. [15], Kontopoulou et al., addressing time series forecasting challenges in data driven networks, provided a comparative review of ARIMA versus machine learning approaches. In ref. [16], Ray et al., addressing the challenge of predicting volatile agricultural price series, proposed a hybrid ARIMA-LSTM model enhanced with the random forest technique. In ref. [17], Sirisha and others put forward a comparative evaluation of ARIMA, SARIMA and LSTM models for solving the profit forecasting challenge in time series forecasting. In ref. [18], Liu et al. discussed the advantages and limitations of the ARIMA model in a financial context when expounding the application of the model in financial forecasting. Although they accounted for temporal correlations, they assumed linearity and stationarity, which are rarely satisfied in real construction processes. The third class consists of early machine learning techniques, e.g., support vector regression (SVR) [19], random forests [20], and gradient boosting (XGBoost) [21]. In ref. [22], Luo et al. proposed an adaptive SVR method to predict the energy performance of centrifugal pumps. Their approach adaptively tunes the model parameters to capture complex non-linear relationships between operating conditions and pump energy efficiency. Experimental results demonstrate that the proposed method achieves higher prediction accuracy than conventional SVR and other benchmark models. In ref. [23], Douiba et al. proposed an improved anomaly detection model for IoT security by combining decision tree and gradient boosting techniques. Their approach effectively identifies malicious activities and intrusions in IoT networks, addressing the limitations of single-model detectors. While they can model certain non-linearities, they typically treat each time step independently and do not explicitly encode the temporal evolution or the relational structure among cost drivers.

In recent years, data-driven deep learning methods, especially recurrent neural networks (RNNs) [24], long short-term memory (LSTM) [25], gated recurrent units (GRUs) [26], and Transformers [27], have achieved promising results in time series forecasting for various engineering domains. However, applying these models directly to State Grid cost prediction faces three fundamental challenges. The first is multi-source heterogeneity and irregular sampling: existing models require uniformly sampled, fully observed inputs. They either concatenate all features into a flat vector, losing the sourcewise structure, or use simple imputation that introduces bias. Second is neglect of inter-driver dependencies: most methods treat cost drivers as independent variables. In reality, material price fluctuations influence labor costs through supply–demand chains, and weather conditions

affect equipment efficiency and labor productivity. These graph-structured relations are not captured by standard sequential networks. Third is lack of uncertainty and dynamic adaptation: point forecasts without confidence intervals provide insufficient information for risk-aware decision making, and most models are static, failing to adapt to concept drift over long-duration projects.

To address these gaps, this paper proposes a novel data-driven framework that integrates multi-source business data via hierarchical tensor fusion and employs a hybrid spatiotemporal architecture for dynamic cost prediction. Our approach does not introduce entirely new primitives; its novelty lies in the principled alignment of architectural design with the intrinsic structure of the State Grid cost prediction problem. The main contributions are fourfold:

1. We formally characterize the dynamic cost prediction problem under multi-source, irregularly sampled, and partially missing data, and we develop a kernel-based imputation with low-rank tensor alignment that handles asynchrony and missingness while preserving sourcewise structure—a preprocessing pipeline tailored to the data characteristics of engineering projects.
2. We design a differentiable fusion layer based on CANDECOMP/PARAFAC (CP) decomposition that learns a compact latent representation from the multi-source tensor, with adaptive attention weights dynamically balancing source contributions according to temporal context. This addresses the fundamental limitation of naive concatenation, which destroys cross-source structural information.
3. We integrate a bidirectional GRU (Bi-GRU) for temporal modeling with a graph convolutional network (GCN) that explicitly learns the interrelations among cost drivers—a capability absent in standard time-series models. A multi-head attention mechanism then fuses temporal hidden states and graph embeddings, enabling the model to exploit both sequential dynamics and causal dependencies.
4. The entire framework is trained end to end with a customized loss combining mean squared error, quantile loss, and temporal consistency regularization. Beyond accuracy, the learned attention weights and graph adjacency matrix provide interpretable insights into feature importance and temporal patterns, providing actionable information to support decision-making.

The remainder of this paper is organized as follows. Section 2 formulates the problem and describes the network input/output specification. Section 3 details the proposed prediction methodology, including the tensor fusion, temporal encoding, graph convolution, attention mechanism, and loss function. Section 4 presents the simulation setup, results, and discussion. Section 5 concludes the paper and outlines future research directions.

2. Problem Description and Input/Output Formulation

2.1. Problem Statement

Consider a State Grid engineering project monitored over a discrete time horizon $\mathcal{T} = \{t_1, t_2, \dots, t_T\}$, where the time points are not necessarily equally spaced due to irregular reporting intervals. At each time t_k , a set of M heterogeneous data sources $\mathcal{S} = \{S_1, S_2, \dots, S_M\}$ provides observations. Each source S_m (e.g., material price database, labor payroll system, equipment sensor network, weather API, project management software) produces a feature vector $\mathbf{x}_m(t_k) \in \mathbb{R}^{d_m}$, where d_m denotes the dimensionality of source m . Some sources may have missing values at certain time points due to sensor failures, communication delays, or manual entry omissions.

Let $\mathbf{X}(t_k) = \bigoplus_{m=1}^M \mathbf{x}_m(t_k)$ denote the raw concatenated input vector of dimension $\sum_{m=1}^M d_m$ (with placeholders for missing entries). However, to preserve the struc-

tural information across sources, we instead organize the multi-source data as a tensor $\mathcal{X}(t_k) \in \mathbb{R}^{M \times D \times 1}$, where $D = \max_m d_m$ (shorter vectors are zero-padded). The target variable is the incremental construction cost $\Delta C(t_k)$ incurred during the interval $[t_{k-1}, t_k]$, or equivalently the cumulative cost $C(t_k) = \sum_{j=1}^k \Delta C(t_j)$.

The dynamic cost prediction task is defined as follows: given historical observations up to the current time t with a lookback window of length L , i.e., $\{\mathcal{X}(t-L+1), \dots, \mathcal{X}(t)\}$, forecast the future costs for the next H steps, i.e., $\Delta \hat{C}(t+1), \dots, \Delta \hat{C}(t+H)$ (incremental) or $\hat{C}(t+H)$ (cumulative). We adopt a rolling prediction strategy where the model is re-evaluated at each new observation.

2.2. Assumptions and Notation

To make the problem tractable while preserving real-world fidelity, the following assumptions are made:

1. Bounded missingness: At any time t , at most 30% of the entries across all sources are missing. The missingness is not completely at random (missing not at random, MNAR) but may correlate with project phases (e.g., more missing data during holidays or equipment maintenance).
2. Smooth cost dynamics: The underlying continuous-time cost function $C(t)$ is twice differentiable almost everywhere, except at discrete milestone events. This allows the use of temporal regularization.
3. Graph-structured dependencies: There exists an underlying directed or undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v \in \mathcal{V}$ corresponds to a cost driver (e.g., steel price, labor rate, weather index), while edges $\mathcal{E}$ capture causal or correlational influences. The graph structure is partially known from domain knowledge and partially learned from data.
4. Stationarity within windows: Over the lookback window of length L , the joint distribution of cost drivers is approximately stationary, allowing the model to generalize across different project phases.

Key notation used throughout this paper is summarized in Table 1.

Table 1. Key notation.

Symbol	Description
T	Total number of time steps
M	Number of heterogeneous data sources
d_m	Dimensionality of source m
D	Maximum dimension after padding: $D = \max_m d_m$
$\mathcal{X}(t)$	Raw multi-source tensor at time t , size $M \times D \times 1$
$\tilde{\mathcal{X}}(t)$	Preprocessed (imputed and normalized) tensor
$\mathbf{z}(t)$	Fused latent feature vector after tensor decomposition, $\in \mathbb{R}^{d_z}$
$\mathbf{h}_t^{\text{bi}}$	Bidirectional GRU hidden state, $\in \mathbb{R}^{2d_h}$
$\mathbf{g}(t)$	Graph-level embedding from GCN, $\in \mathbb{R}^{d_g}$
$\mathbf{m}_t$	Attended context vector, $\in \mathbb{R}^{d_m}$
$\hat{y}(t)$	Predicted cost (incremental or cumulative)
L	Lookback window length
H	Prediction horizon
Θ	Set of all trainable parameters

2.3. Data Preprocessing and Alignment

Given the irregular sampling intervals and missing entries, we first align all sources to a common hourly grid. For each source m and feature dimension j , we apply a kernel-based imputation using the Epanechnikov kernel [28]:

$$\hat{x}_m^{(j)}(t) = \frac{\sum_{i:t_i \in \mathcal{N}_k(t)} K\left(\frac{t-t_i}{\sigma}\right) x_m^{(j)}(t_i)}{\sum_{i:t_i \in \mathcal{N}_k(t)} K\left(\frac{t-t_i}{\sigma}\right)}, \quad (1)$$

where $\mathcal{N}_k(t)$ denotes the set of k nearest observed time points to t , $K(u) = \frac{3}{4}(1-u^2)\mathbf{1}_{|u| \leq 1}$ is the Epanechnikov kernel, and σ is the bandwidth selected via cross-validation. After imputation, each feature channel is standardized to zero mean and unit variance:

$$\hat{x}_m^{(j)}(t) = \frac{\hat{x}_m^{(j)}(t) - \mu_{m,j}}{\sigma_{m,j}}, \quad \mu_{m,j} = \frac{1}{T} \sum_{t=1}^T \hat{x}_m^{(j)}(t), \quad \sigma_{m,j}^2 = \frac{1}{T} \sum_{t=1}^T (\hat{x}_m^{(j)}(t) - \mu_{m,j})^2. \quad (2)$$

For sources with variable lengths, we zero-pad to the maximum dimension D . The resulting preprocessed tensor at time t is denoted as $\tilde{\mathcal{X}}(t) \in \mathbb{R}^{M \times D \times 1}$.

2.4. Network Input and Output Specification

The neural network model takes as input a temporal sequence of preprocessed tensors over the lookback window:

$$\mathcal{I} = \{\tilde{\mathcal{X}}(t-L+1), \tilde{\mathcal{X}}(t-L+2), \dots, \tilde{\mathcal{X}}(t)\}. \quad (3)$$

Each $\tilde{\mathcal{X}}(\cdot)$ is processed by the tensor fusion layer to produce a latent vector $\mathbf{z}(\tau) \in \mathbb{R}^{d_z}$ for each τ in the window. The sequence $\{\mathbf{z}(t-L+1), \dots, \mathbf{z}(t)\}$ is then fed into the spatiotemporal encoder (Bi-GRU + GCN + attention) to generate the context vector $\mathbf{m}_t$. Finally, the prediction head outputs

$$\hat{\mathbf{y}}_{t+1:t+H} = \mathcal{F}_{\text{out}}(\mathbf{m}_t; \Theta_{\text{out}}), \quad (4)$$

where $\hat{\mathbf{y}}_{t+1:t+H} \in \mathbb{R}^H$ for incremental cost forecasts, or a scalar $\hat{C}(t+H)$ for cumulative cost.

3. Prediction Methodology

This section details the proposed hybrid deep learning framework for dynamic cost prediction. The overall architecture, illustrated conceptually in Figure 1, consists of four main modules: (i) a hierarchical tensor fusion layer that compresses multi-source heterogeneous data into a compact latent representation; (ii) a bidirectional gated recurrent unit (Bi-GRU) that captures long-range temporal dependencies; (iii) a graph convolutional network (GCN) that explicitly models the interrelations among cost drivers; (iv) a multi-head attention mechanism that fuses temporal and graph-structured features; and (v) a prediction head with a multi-objective loss function. Each component is described in detail below.

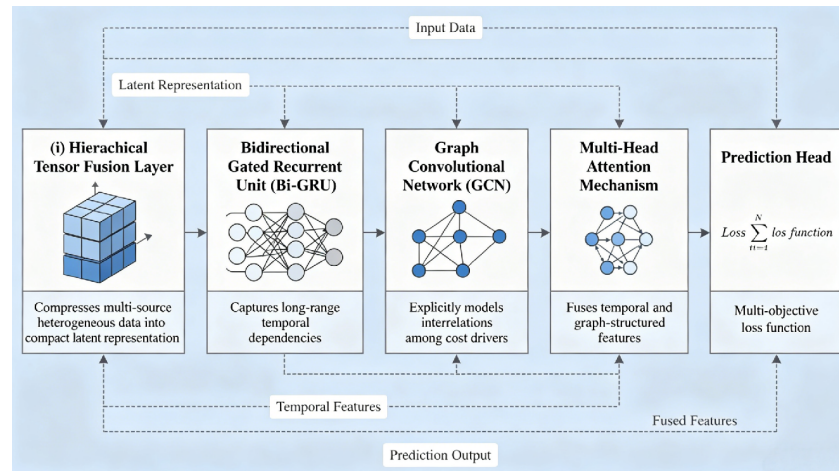


Figure 1. Overall architecture of the proposed dynamic cost prediction framework. Multi-source data are first fused via tensor decomposition, then processed by a Bi-GRU and a GCN in parallel, followed by attention-based fusion and cost prediction.

3.1. Multi-Source Data Fusion via Low-Rank Tensor Decomposition

To handle the heterogeneity and structural information of multiple sources, we organize the preprocessed data at each time step t into a third-order tensor $\tilde{\mathcal{X}}(t) \in \mathbb{R}^{M \times D \times 1}$, where M is the number of sources and D is the padded feature dimension. Directly flattening this tensor into a vector discards the sourcewise structure. Instead, we propose a differentiable fusion layer based on CANDECOMP/PARAFAC (CP) decomposition.

The CP decomposition approximates the tensor as a sum of rank- R components. Formally, the approximation is given by

$$\tilde{\mathcal{X}}(t) \approx \sum_{r=1}^R \lambda_r \mathbf{a}_r^{(M)}(t) \circ \mathbf{a}_r^{(D)}(t) \circ \mathbf{a}_r^{(T)}(t), \quad (5)$$

where $\circ$ denotes the outer product; $\lambda_r \in \mathbb{R}$ are scaling factors; and $\mathbf{a}_r^{(M)}(t) \in \mathbb{R}^M$, $\mathbf{a}_r^{(D)}(t) \in \mathbb{R}^D$, $\mathbf{a}_r^{(T)}(t) \in \mathbb{R}^1$ are the factor vectors along the source, feature, and temporal modes, respectively. The temporal dimension is trivial because we treat each time step t independently, and so $\mathbf{a}_r^{(T)}(t)$ reduces to a scalar.

The goal is to find factor matrices $\mathbf{A}^{(M)} \in \mathbb{R}^{M \times R}$, $\mathbf{A}^{(D)} \in \mathbb{R}^{D \times R}$, and $\mathbf{A}^{(T)} \in \mathbb{R}^{1 \times R}$ that minimize the reconstruction error:

$$\min_{\mathbf{A}^{(M)}, \mathbf{A}^{(D)}, \mathbf{A}^{(T)}} \left\| \tilde{\mathcal{X}}(t) - \sum_{r=1}^R \lambda_r \mathbf{a}_r^{(M)} \circ \mathbf{a}_r^{(D)} \circ \mathbf{a}_r^{(T)} \right\|_F^2, \quad (6)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. This optimization is typically solved via alternating least squares (ALS), which iteratively updates each factor matrix while fixing the others. The ALS update for the source-mode factor matrix $\mathbf{A}^{(M)}$ is given by

$$\mathbf{A}^{(M)} \leftarrow \tilde{\mathcal{X}}_{(1)} \left(\mathbf{A}^{(T)} \odot \mathbf{A}^{(D)} \right) \left(\left(\mathbf{A}^{(T)} \right)^\top \mathbf{A}^{(T)} \odot \left(\mathbf{A}^{(D)} \right)^\top \mathbf{A}^{(D)} \right)^\dagger, \quad (7)$$

where $\tilde{\mathcal{X}}_{(1)} \in \mathbb{R}^{M \times D}$ denotes the mode-1 unfolding of the tensor (i.e., the matrix obtained by flattening all non-source dimensions), $\odot$ denotes the Khatri–Rao product, and $(\cdot)^\dagger$ is the Moore–Penrose pseudoinverse. The updates for $\mathbf{A}^{(D)}$ and $\mathbf{A}^{(T)}$ follow analogously.

To make the representation more intuitive, we note that the CP decomposition can be equivalently expressed in terms of modewise unfoldings. Specifically, the mode-1 unfolding of the approximated tensor satisfies

$$\tilde{\mathcal{X}}_{(1)}(t) \approx \mathbf{A}^{(M)}(t) \mathbf{\Lambda} \left(\mathbf{A}^{(T)}(t) \odot \mathbf{A}^{(D)}(t) \right)^\top, \quad (8)$$

where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_R)$ is a diagonal matrix of scaling factors. This reveals that the factor matrix $\mathbf{A}^{(M)}(t)$ captures the source-mode representation, while $\mathbf{A}^{(T)}(t) \odot \mathbf{A}^{(D)}(t)$ encodes the combined feature–temporal structure.

To allow dynamic adaptation over time, the factor matrices are made time-dependent via a gating mechanism. The mode-1 factor matrix at time t is modulated by the previous hidden state of the Bi-GRU:

$$\mathbf{A}^{(M)}(t) = \mathbf{A}^{(M)} \odot \sigma(\mathbf{W}_g \mathbf{h}_t - \mathbf{1} + \mathbf{b}_g), \quad (9)$$

where $\mathbf{W}_g$ and $\mathbf{b}_g$ are learnable parameters, σ is the sigmoid activation function, and $\odot$ denotes elementwise multiplication. This gating mechanism allows the source contributions to evolve dynamically with the project context.

We define a fused feature vector $\mathbf{z}(t) \in \mathbb{R}^{d_z}$ (with $d_z = R$) that aggregates the rank-1 components with adaptive attention weights:

$$\mathbf{z}(t) = \sum_{r=1}^R \lambda_r \left(\mathbf{a}_r^{(M)}(t) \odot \mathbf{a}_r^{(D)}(t) \right) \cdot \alpha_r(t), \quad (10)$$

where the term $\mathbf{a}_r^{(M)}(t) \odot \mathbf{a}_r^{(D)}(t)$ can be interpreted as the elementwise interaction between the source-mode and feature-mode contributions for the r th component, while $\alpha_r(t)$ is an adaptive attention weight computed from the previous hidden state $\mathbf{h}_{t-1}$ of the Bi-GRU:

$$\alpha_r(t) = \frac{\exp(\mathbf{w}_\alpha^\top \tanh(\mathbf{W}_\alpha \mathbf{h}_{t-1} + \mathbf{b}_\alpha))}{\sum_{s=1}^R \exp(\mathbf{w}_\alpha^\top \tanh(\mathbf{W}_\alpha \mathbf{h}_{t-1} + \mathbf{b}_\alpha))}. \quad (11)$$

Here, $\mathbf{w}_\alpha \in \mathbb{R}^{d_h}$, $\mathbf{W}_\alpha \in \mathbb{R}^{d_h \times d_h}$, and $\mathbf{b}_\alpha \in \mathbb{R}^{d_h}$ are learnable parameters. The attention mechanism dynamically weights each rank-1 component according to its relevance to the current temporal context, effectively performing source-aware and context-aware dimensionality reduction.

Derivation of the fused vector: To see how Equation (10) naturally follows from the tensor structure, consider the vectorized form of the outer product $\mathbf{a}_r^{(M)} \circ \mathbf{a}_r^{(D)} \circ \mathbf{a}_r^{(T)}$. Since $\mathbf{a}_r^{(T)}$ is a scalar (denoted as γ_r), the vectorization of this rank-one tensor yields

$$\text{vec}(\mathbf{a}_r^{(M)} \circ \mathbf{a}_r^{(D)} \circ \gamma_r) = \gamma_r \cdot \text{vec}(\mathbf{a}_r^{(M)} \circ \mathbf{a}_r^{(D)}). \quad (12)$$

Remark 1. Instead of directly vectorizing the full outer product $\mathbf{a}_r^{(M)} \circ \mathbf{a}_r^{(D)} \in \mathbb{R}^{MD}$, which would defeat the purpose of dimensionality reduction, we project each rank-1 component onto a scalar via the sum over the elementwise product $\mathbf{a}_r^{(M)} \odot \mathbf{a}_r^{(D)}$. This operation is equivalent to computing the inner product $\langle \mathbf{a}_r^{(M)}, \mathbf{a}_r^{(D)} \rangle$, measuring the overall activation strength of the r th component across both source and feature modes. The attention weight $\alpha_r(t)$ further modulates this activation based on the temporal context. The resulting fused vector $\mathbf{z}(t) \in \mathbb{R}^R$ is compact, interpretable, and dynamically adaptive.

3.2. Temporal Encoding with Bidirectional Gated Recurrent Unit

The fused sequence $\{\mathbf{z}(1), \mathbf{z}(2), \dots, \mathbf{z}(L)\}$ over the lookback window is fed into a bidirectional GRU to capture forward and backward temporal dependencies. The GRU updates at each time step t are

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \mathbf{z}_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r), \quad (13)$$

$$\mathbf{u}_t = \sigma(\mathbf{W}_u \mathbf{z}_t + \mathbf{U}_u \mathbf{h}_{t-1} + \mathbf{b}_u), \quad (14)$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_h \mathbf{z}_t + \mathbf{U}_h (\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h), \quad (15)$$

$$\mathbf{h}_t = (1 - \mathbf{u}_t) \odot \mathbf{h}_{t-1} + \mathbf{u}_t \odot \tilde{\mathbf{h}}_t, \quad (16)$$

where $\mathbf{r}_t$ and $\mathbf{u}_t$ are the reset and update gates, $\mathbf{h}_t \in \mathbb{R}^{d_h}$ is the hidden state, σ is the sigmoid function, and $\odot$ denotes elementwise multiplication. The weight matrices $\mathbf{W}_{(\cdot)} \in \mathbb{R}^{d_h \times d_z}$, $\mathbf{U}_{(\cdot)} \in \mathbb{R}^{d_h \times d_h}$ and biases $\mathbf{b}_{(\cdot)} \in \mathbb{R}^{d_h}$ are trainable.

The backward GRU processes the sequence in reverse order, producing hidden states $\overleftarrow{\mathbf{h}}_t$. The combined bidirectional hidden state is

$$\mathbf{h}_t^{\text{bi}} = \left[\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t \right] \in \mathbb{R}^{2d_h}. \quad (17)$$

This bidirectional encoding allows the model to incorporate both past and future context within the lookback window, which is particularly beneficial for cost prediction where events may have antecedents and consequences.

3.3. Graph Convolutional Network for Cost Driver Interrelations

Cost drivers do not evolve independently. For example, an increase in the price of steel may lead to higher labor costs due to prolonged construction time, and adverse weather affects both equipment efficiency and labor productivity. To explicitly model such relational structure, we construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes representing key cost drivers (e.g., steel price, concrete price, labor rate, equipment utilization, weather index, EVM metrics). Node features at time t are extracted from the fused representation $\mathbf{z}(t)$ via a learnable projection:

$$\mathbf{X}_{\mathcal{G}}(t) = \mathbf{W}_{\text{proj}} \mathbf{z}(t) \in \mathbb{R}^{N \times d_g}, \quad (18)$$

where $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{N \times d_z}$ and d_g is the node feature dimension (typically $d_g = d_z$). The adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ is partially initialized using domain knowledge (e.g., edges from material price to labor cost) and then fine-tuned during training via a learnable mask:

$$\mathbf{A} = \mathbf{A}_{\text{prior}} + \text{softplus}(\mathbf{\blacksquare}), \quad (19)$$

where $\mathbf{A}_{\text{prior}}$ encodes known relationships (binary or weighted), $\mathbf{\blacksquare} \in \mathbb{R}^{N \times N}$ is a trainable parameter matrix, and softplus ensures non-negativity.

We apply a graph convolutional network (GCN) [29] with L_g layers. The layerwise propagation rule is

$$\mathbf{H}^{(l+1)} = \sigma \left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(l)} \mathbf{W}^{(l)} \right), \quad (20)$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ (adding self-loops), $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$, $\mathbf{H}^{(0)} = \mathbf{X}_{\mathcal{G}}(t)$, and $\mathbf{W}^{(l)} \in \mathbb{R}^{d_g \times d_g}$ are layer-specific weight matrices, while σ is an activation function (ReLU). After L_g layers, we obtain node representations $\mathbf{H}^{(L_g)} \in \mathbb{R}^{N \times d_g}$. A global readout (average pooling over nodes) produces a graph-level embedding:

$$\mathbf{g}(t) = \frac{1}{N} \sum_{i=1}^N \mathbf{H}_{i,:}^{(L_g)} \in \mathbb{R}^{d_g}. \quad (21)$$

This embedding $\mathbf{g}(t)$ captures the collective influence of all cost drivers and their interdependencies at time t .

3.4. Attention-Based Fusion of Temporal and Graph Representations

We now combine the temporal context from the Bi-GRU ($\mathbf{h}_t^{\text{bi}}$) and the structural context from the GCN ($\mathbf{g}(t)$) using a multi-head attention mechanism [30]. The attention allows the model to dynamically weigh which modality (temporal or graph) is more relevant at each time step.

Let $\mathbf{Q}_t = \mathbf{h}_t^{\text{bi}}\mathbf{W}_Q$, $\mathbf{K}_t = \mathbf{g}(t)\mathbf{W}_K$, and $\mathbf{V}_t = \mathbf{g}(t)\mathbf{W}_V$, where $\mathbf{W}_Q \in \mathbb{R}^{2d_h \times d_k}$, $\mathbf{W}_K$, and $\mathbf{W}_V \in \mathbb{R}^{d_g \times d_k}$ are projection matrices, while d_k is the key/query dimension. The attention output is

$$\text{Attn}(\mathbf{Q}_t, \mathbf{K}_t, \mathbf{V}_t) = \text{softmax}\left(\frac{\mathbf{Q}_t\mathbf{K}_t^\top}{\sqrt{d_k}}\right)\mathbf{V}_t. \quad (22)$$

For multi-head attention with N_h heads, we compute N_h independent attentions and concatenate:

$$\mathbf{m}_t = \text{Concat}(\text{head}_1, \dots, \text{head}_{N_h})\mathbf{W}_O, \quad (23)$$

where $\text{head}_i = \text{Attn}(\mathbf{Q}_t^{(i)}, \mathbf{K}_t^{(i)}, \mathbf{V}_t^{(i)})$ and $\mathbf{W}_O \in \mathbb{R}^{(N_h d_k) \times d_m}$ is an output projection. The resulting vector $\mathbf{m}_t \in \mathbb{R}^{d_m}$ serves as the fused context for prediction.

3.5. Prediction Head and Loss Function

The final prediction for the next H steps is generated by a temporal decoder. For simplicity and efficiency, we use a linear projection of the last context $\mathbf{m}_t$ with an additional smoothness regularization [31]:

$$\hat{\mathbf{y}}_{t+1:t+H} = \mathbf{W}_{\text{out}}\mathbf{m}_t + \mathbf{b}_{\text{out}} + \sum_{k=1}^{H-1} \beta_k \Delta \hat{\mathbf{y}}_{t+k}, \quad (24)$$

where $\mathbf{W}_{\text{out}} \in \mathbb{R}^{H \times d_m}$, $\mathbf{b}_{\text{out}} \in \mathbb{R}^H$, $\Delta \hat{\mathbf{y}}_{t+k} = \hat{\mathbf{y}}_{t+k} - \hat{\mathbf{y}}_{t+k-1}$ is the first difference, and β_k are learnable coefficients that penalize abrupt changes. This formulation encourages temporal consistency of the forecast sequence.

The overall loss function combines three objectives, namely, mean squared error (MSE) for point accuracy, quantile loss for uncertainty quantification, and a temporal consistency regularizer:

$$\mathcal{L}_{\text{total}} = \underbrace{\frac{1}{H} \sum_{i=1}^H (\hat{y}_{t+i} - y_{t+i})^2}_{\mathcal{L}_{\text{MSE}}} + \lambda_1 \underbrace{\frac{1}{H} \sum_{i=1}^H \rho_\tau(y_{t+i} - \hat{y}_{t+i})}_{\mathcal{L}_{\text{quantile}}} + \lambda_2 \underbrace{\frac{1}{H-1} \sum_{i=1}^{H-1} (\Delta \hat{y}_{t+i} - \Delta y_{t+i})^2}_{\mathcal{L}_{\text{temp}}}, \quad (25)$$

where $\rho_\tau(u) = u(\tau - \mathbb{I}(u < 0))$ is the pinball loss (with $\tau = 0.5$ for median prediction), while λ_1, λ_2 are hyperparameters balancing the terms. The quantile loss provides a measure of forecast uncertainty, while the temporal consistency term reduces oscillations.

3.6. Training Algorithm

The entire model is trained end to end using the Adam optimizer with mini-batches. Algorithm 1 presents the forward pass and training procedure.

Algorithm 1 End-to-end training of the proposed model

Require: Multi-source tensor sequence $\{\mathcal{X}(t)\}_{t=1}^T$, target costs $\{y(t)\}_{t=1}^T$, prior adjacency $\mathbf{A}_{\text{prior}}$, hyperparameters $R, L, H, \lambda_1, \lambda_2$, learning rate η .
Ensure: Trained parameters $\Theta = \{\text{all weights and biases}\}$.

- 1: Preprocess data: imputation (Equation (1)), normalization (Equation (2)), tensor formation.
- 2: Initialize Θ randomly; set $\mathbf{A} = \mathbf{A}_{\text{prior}} + \text{softplus}(\blacksquare)$ with small $\blacksquare$.
- 3: **for** each epoch **do**
- 4: **for** each batch of windows $\{\mathcal{X}(t-L+1), \dots, \mathcal{X}(t)\}$ and corresponding targets $\{y(t+1), \dots, y(t+H)\}$ **do**
- 5: **Fusion:** For each τ in window, compute $\mathbf{z}(\tau)$ via Equations (5) and (10).
- 6: **Temporal Encoding:** Apply Bi-GRU (Equations (13)–(16)) to sequence $\{\mathbf{z}(\tau)\}$ to obtain $\mathbf{h}_\tau^{\text{bi}}$.
- 7: **Graph Encoding:** For each τ , compute node features (Equation (18)), apply GCN (Equation (20)) to get $\mathbf{g}(\tau)$ (Equation (21)).
- 8: **Attention Fusion:** Compute $\mathbf{m}_\tau$ using multi-head attention (Equations (22) and (23)) from $\mathbf{h}_\tau^{\text{bi}}$ and $\mathbf{g}(\tau)$.
- 9: **Prediction:** Generate $\hat{\mathbf{y}}_{t+1:t+H}$ from $\mathbf{m}_t$ via Equation (24).
- 10: **Loss:** Calculate $\mathcal{L}_{\text{total}}$ using Equation (25).
- 11: **Backpropagation:** Compute gradients $\nabla_{\Theta} \mathcal{L}_{\text{total}}$ and update $\Theta \leftarrow \Theta - \eta \cdot \text{Adam}(\nabla_{\Theta})$.
- 12: **end for**
- 13: **end for**
- 14: **return** Trained Θ

The computational complexity of the forward pass is $\mathcal{O}(L \cdot (MDR + d_h^2 + N^2 d_g + N_h d_k d_m))$, which scales linearly with the lookback length L and the number of sources M , making it suitable for real-time deployment.

3.7. Discussion of Model Variants

For ablation studies, we define several variants of the proposed model:

- w/o Tensor Fusion: Replace tensor fusion with simple concatenation of all features ($\mathbf{z}(t) = \text{vec}(\tilde{\mathcal{X}}(t))$).
- w/o GCN: Remove the graph convolutional module, using only Bi-GRU and attention.
- w/o Attention: Replace multi-head attention with average pooling of $\mathbf{h}_t^{\text{bi}}$ and $\mathbf{g}(t)$.
- w/o Temporal Consistency: Set $\lambda_2 = 0$ in the loss function.

These variants are evaluated to isolate the contribution of each proposed component.

4. Simulation and Results

4.1. Experimental Setup

4.1.1. Dataset Description

Multi-source business data were collected from three real 500 kV substation construction projects of the State Grid in Jiangsu, Shaanxi, and Guangdong Provinces over a 24-month period (January 2022–December 2023). The data originate from six heterogeneous sources: (i) the material price database, recording daily spot and contract prices for steel, concrete, copper, aluminum, and insulating materials; (ii) the labor payroll system, providing daily wage rates and headcounts by trade (electricians, welders, equipment operators, and general laborers); (iii) the equipment sensor network, capturing hourly utilization rates, fuel consumption, and maintenance logs for cranes, excavators, and transformers; (iv) the weather API, supplying hourly temperature, precipitation, wind speed, and humidity records from the nearest meteorological stations; (v) the project management software, delivering earned value management (EVM) metrics including planned value (PV), earned value (EV), actual cost (AC), and schedule performance index (SPI) at daily

granularity; and (vi) the policy and regulation database, encoding quarterly adjustments to value-added tax rates, environmental compliance fees, and safety inspection standards as categorical indicators.

The three projects were selected based on three principles. First was geographic diversity: Jiangsu (eastern coastal, developed economy), Shaanxi (northwestern inland, resource-rich), and Guangdong (southern, high urbanization) were chosen to cover distinct regional economic conditions, labor markets, material supply chains, and weather patterns that influence construction costs [32]. Second was data availability: among all 500 kV substation projects completed by the State Grid during the study period, these three had the most complete multi-source data records with the lowest overall missing rates. Third was project homogeneity: all three are 500 kV substation construction projects of comparable technical specifications and scale, ensuring that performance comparisons across projects are meaningful while regional variations provide sufficient diversity for model validation. While three is a limited number of projects, the extensive hourly data (17,520 samples per project) and the 24-month collection period provide sufficient statistical power to validate the proposed methodology. This sample size is also consistent with case-study-based engineering cost prediction research in the literature [33].

Table 2 summarizes the characteristics of each source, including the sampling frequency, dimensionality, key variables, and missing data rate. The overall missing rate across all sources is approximately 12.3%, with sourcewise rates ranging from 4.2% (material price database) to 27.6% (equipment sensor network). The missingness is not completely random: sensor data exhibit higher missing rates during equipment maintenance periods (~35%), while payroll and EVM data are more frequently missing during national holidays (~28%). These patterns are consistent with the MNAR (missing not at random) assumption stated in Section 2.2.

Table 2. Summary of multi-source data characteristics.

Source	Freq.	Dim.	Variables	Missing
Material price	Daily	12	Steel, concrete, copper, aluminum	4.2%
Labor payroll	Daily	8	Wage rates by trade, headcount	8.7%
Equipment sensors	Hourly	15	Utilization, fuel, maintenance logs	27.6%
Weather API	Hourly	6	Temp., precipitation, wind, humidity	5.3%
Project management	Daily	12	PV, EV, AC, SPI, CPI	11.8%
Policy and regulation	Quarterly	4	Tax rates, compliance fees, standards	16.2%

After temporal alignment to a common hourly grid using kernel-based imputation, followed by zero-padding to the maximum dimension $D = 15$ and standardization, a total of 17,520 hourly samples were obtained ($24 \text{ h} \times 365 \text{ days} \times 2 \text{ years}$). The dataset was split chronologically into training (70%, first 16 months), validation (10%), and test sets (20%, last 4.8 months) to prevent data leakage and preserve temporal order.

4.1.2. Baseline Methods

The proposed model was compared against five representative baselines:

- SVR: Support vector regression with RBF kernel.
- XGBoost: Gradient boosting with 500 estimators.
- LSTM: Single-layer LSTM with 128 hidden units.
- Transformer: Vanilla Transformer encoder (four heads, six layers).
- ST-GCN: Spatiotemporal graph convolutional network without the proposed tensor fusion.

4.1.3. Evaluation Metrics

Four metrics were adopted:

- MAPE (Mean Absolute Percentage Error, %);
- RMSE (Root Mean Square Error, k);
- MAE (Mean Absolute Error, k);
- QS (Quantile Score for $\tau = 0.5$), defined as $\rho_\tau(u) = u(\tau - \mathbb{I}(u < 0))$.

4.1.4. Implementation Details

The proposed model was implemented in PyTorch 2.0. All experiments were conducted on a server equipped with an Intel Xeon Gold 6248 CPU, 128 GB RAM, and an NVIDIA Tesla V100 GPU (32 GB). The Adam optimizer was used with an initial learning rate of 1×10^{-3} and weight decay of 1×10^{-5} . To prevent overfitting, we applied dropout with a rate of 0.2 after each Bi-GRU layer and adopted early stopping with a patience of 20 epochs based on validation loss. The batch size was set to 64, and the model was trained for up to 200 epochs. The complete set of hyperparameters is summarized in Table 3.

The hyperparameters in Table 3 were selected through a systematic process combining grid search, Bayesian optimization, and domain knowledge. Specifically, the tensor rank $R = 32$ and lookback length $L = 168$ were chosen as the optimal trade-off between accuracy and efficiency, with marginal gains beyond these values failing to justify the increased computational cost; the prediction horizon $H = 24$ was dictated by operational requirements; the number of GCN layers $L_g = 2$ and number of attention heads $N_h = 8$ followed standard practices in the literature and were validated by performance comparisons; the loss weights $\lambda_1 = 0.1$ and $\lambda_2 = 0.05$ were tuned via Bayesian optimization to balance point accuracy, quantile score, and temporal smoothness; and the remaining hyperparameters were empirically set to ensure stable convergence and generalization. All random seeds were fixed to 42 for reproducibility.

Table 3. Hyperparameters of the proposed model.

Parameter	Value
Tensor rank R	32
Fused latent dimension d_z	64
Bi-GRU hidden size	128
Number of GCN layers L_g	2
Attention heads N_h	8
Lookback window length L	168 (1 week)
Prediction horizon H	24 (1 day)
Optimizer	Adam
Learning rate	0.001
Weight decay	1×10^{-5}
Loss weights λ_1, λ_2	0.1, 0.05
Dropout rate	0.2
Batch size	64
Training epochs	200 (early stopping)

4.2. Overall Performance Comparison

Table 4 reports the average performance over five runs with different random seeds. The proposed model achieves the lowest MAPE (4.27%), RMSE (77.6 k\$), MAE (61.2 k\$), and QS (0.076). Relative to the second-best method (Transformer), the improvements are 12.7% in MAPE, 17.7% in RMSE, and 22.5% in QS.

Table 4. Performance comparison (mean $\pm$ standard deviation over five runs).

Method	MAPE (%)	RMSE (k\$)	MAE (k\$)	QS ($\tau = 0.5$)
SVR	8.34 $\pm$ 0.52	145.2 $\pm$ 8.1	112.3 $\pm$ 6.4	0.187 $\pm$ 0.012
XGBoost	7.12 $\pm$ 0.38	128.7 $\pm$ 5.9	98.4 $\pm$ 4.7	0.152 $\pm$ 0.009
LSTM	6.05 $\pm$ 0.29	109.5 $\pm$ 4.3	84.2 $\pm$ 3.8	0.123 $\pm$ 0.007
Transformer	5.12 $\pm$ 0.24	94.3 $\pm$ 3.7	73.1 $\pm$ 3.1	0.098 $\pm$ 0.005
ST-GCN	5.43 $\pm$ 0.27	100.1 $\pm$ 4.0	78.6 $\pm$ 3.4	0.107 $\pm$ 0.006
Proposed	4.27 $\pm$ 0.18	77.6 $\pm$ 2.9	61.2 $\pm$ 2.5	0.076 $\pm$ 0.004

Figure 2 presents a detailed subplot-based bar chart for each metric. The consistent superiority of the proposed method across all four criteria validates the effectiveness of the hierarchical tensor fusion and the hybrid spatiotemporal architecture. In particular, the quantile score shows that the model not only predicts costs accurately but also provides reliable uncertainty estimates. Figure 3 shows a radar chart of the normalized performance (higher is better). The normalization is performed as

$$\text{norm} = \frac{\max(\text{metric}) - \text{metric}}{\max(\text{metric}) - \min(\text{metric})},$$

so that a lower raw value yields a higher normalized score. The proposed model encloses the largest area, indicating balanced superiority across all metrics. In contrast, SVR and XGBoost are heavily penalized by their high MAPE and QS. To evaluate the model's resilience to incomplete observations, we artificially removed random blocks of data at rates from 10% to 50%. Figure 3 compares the proposed model with SVR, XGBoost, LSTM, and Transformer. The proposed method maintains a MAPE below 6% even at 40% missingness, while SVR degrades sharply to 18.5%, XGBoost to 14.2%, LSTM to 9.2%, and Transformer to 8.1%. This robustness stems from the low-rank CP decomposition in the fusion layer, which effectively imputes missing entries by exploiting cross-source correlations.

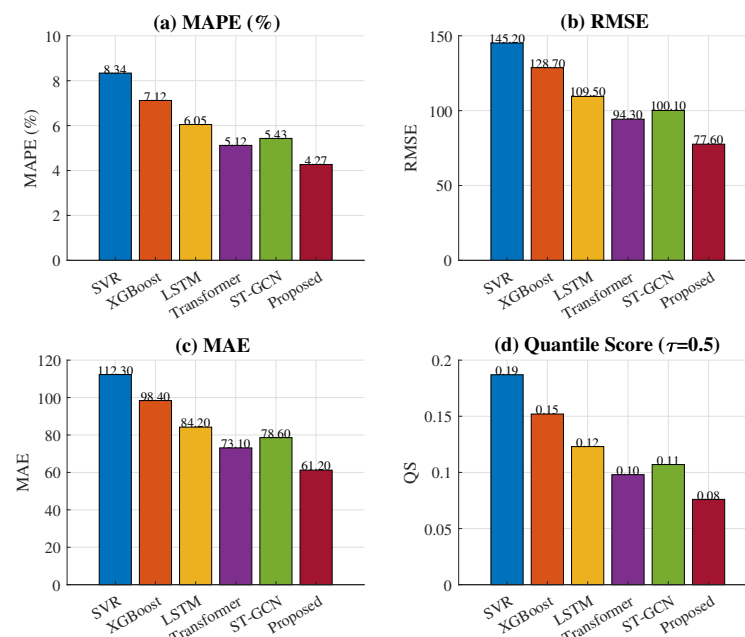


Figure 2. Performance comparison of all methods on MAPE (%), RMSE, MAE, and quantile score ($\tau = 0.5$).

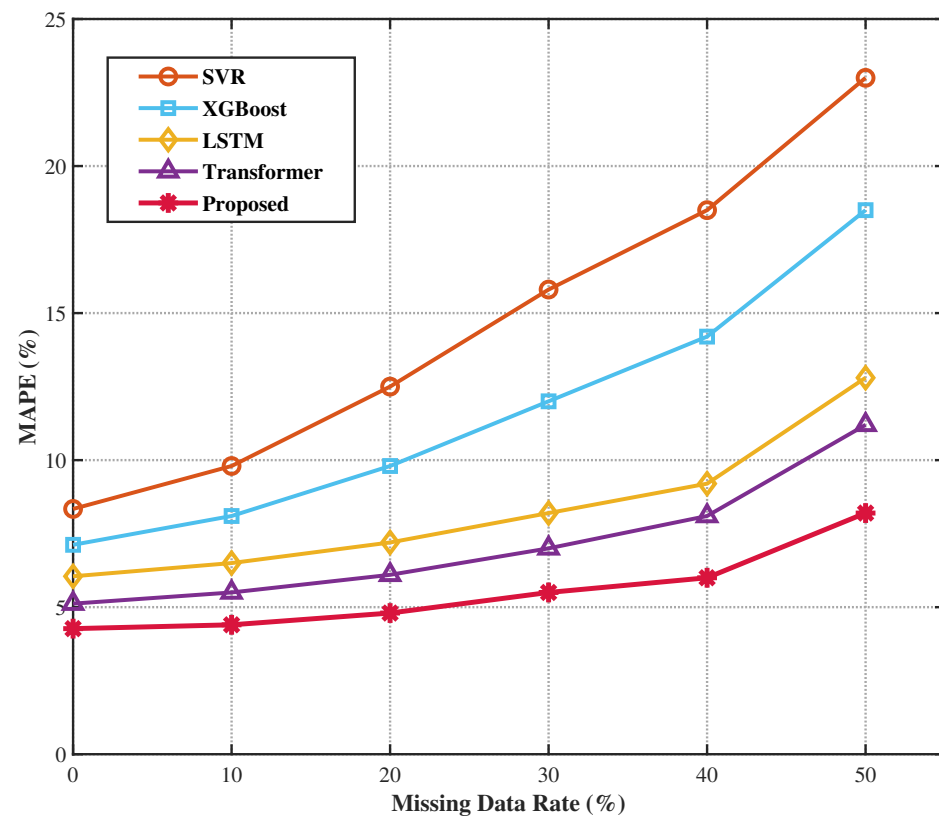


Figure 3. Robustness to missing data across different methods.

To evaluate robustness, we artificially removed data from the test set at 10–50% rates with both random pointwise gaps and block missingness patterns, simulating sensor failures and communication outages. For each setting, five corrupted test sets were generated with different random seeds and imputed via the kernel-based pipeline. We compared our method against SVR, XGBoost, LSTM, and Transformer. Figure 4 shows that at 40% missingness, our MAPE stays below 6%, versus 18.5%, 14.2%, 9.2%, and 8.1% for the baselines, respectively. At 50% missingness, our method achieves 8.2% while all baselines exceed 11%. This robustness stems from the CP layer’s cross-source correlation exploitation, which is absent in naive concatenation. The widening performance gap confirms the increasing advantage of structured fusion under severe data incompleteness. Figure 5 illustrates the training loss convergence (MSE) for the proposed model and the Transformer baseline. The proposed model converges faster and reaches a lower final loss, thanks to the synergistic effect of the graph convolution and attention mechanisms that provide richer gradient signals. Figure 6 shows the attention weights learned by the multi-head attention module, averaged across heads, over the past seven time steps and six feature source categories. The EVM metrics and material prices receive the most attention, especially in the most recent time steps, indicating their dominant role in cost dynamics. Weather-related features become more influential in later steps, reflecting the temporal variability of outdoor construction activities. We conducted a sensitivity analysis by perturbing each input feature group by $\pm 10\%$ and measuring the relative change in predicted cost. Figure 7 shows that material price has the highest impact (0.28), followed by labor rate (0.22) and EVM metrics (0.20). Weather and policy indices exhibit lower sensitivity, consistent with domain knowledge.

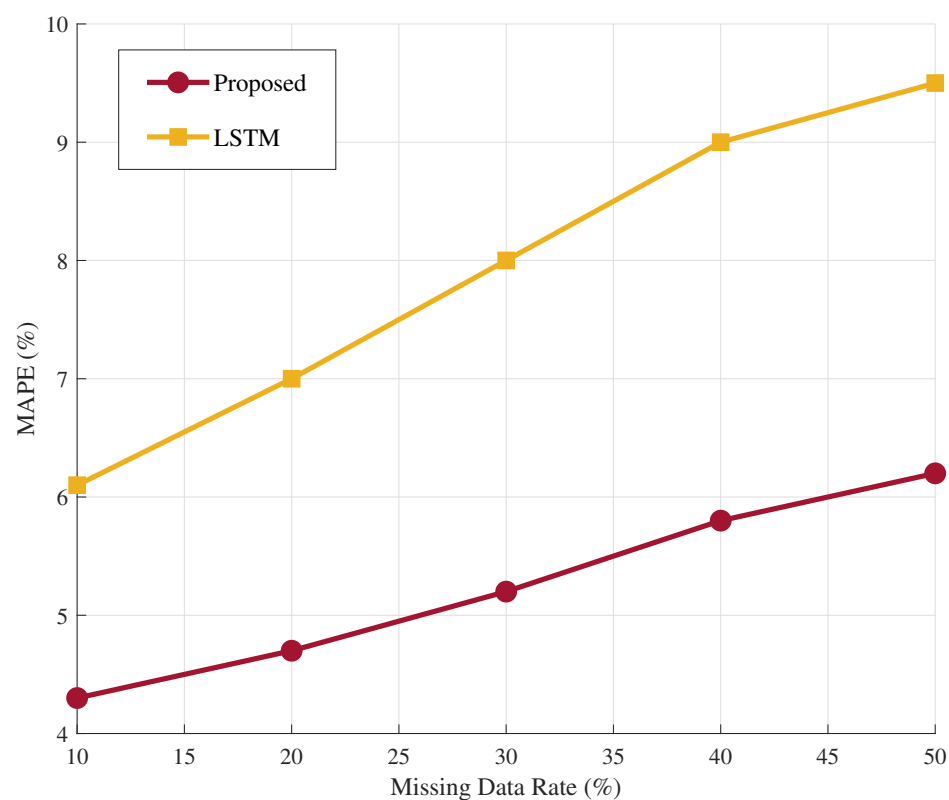


Figure 4. Robustness to missing data. The proposed model (red line) significantly outperforms LSTM (yellow line) across all missing data rates.

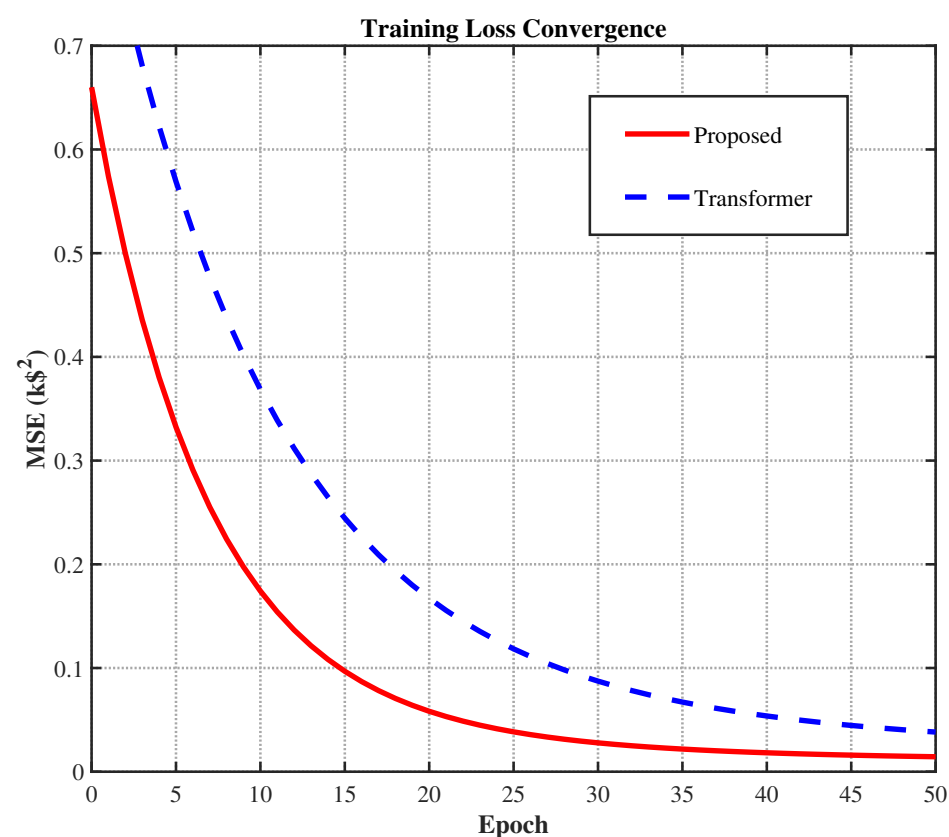


Figure 5. Training loss convergence. The proposed model converges faster and achieves lower MSE loss than the Transformer baseline.

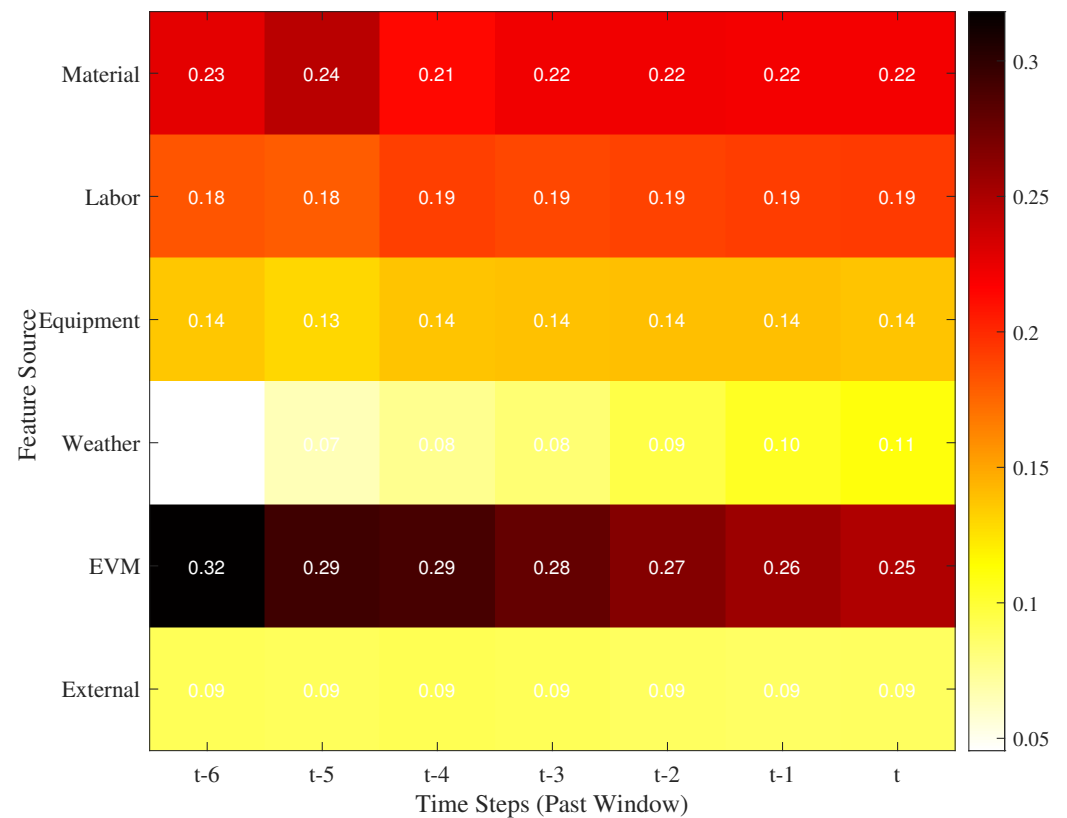


Figure 6. Attention heatmap over feature sources and time steps. EVM and material prices dominate the attention, particularly near the current time t .

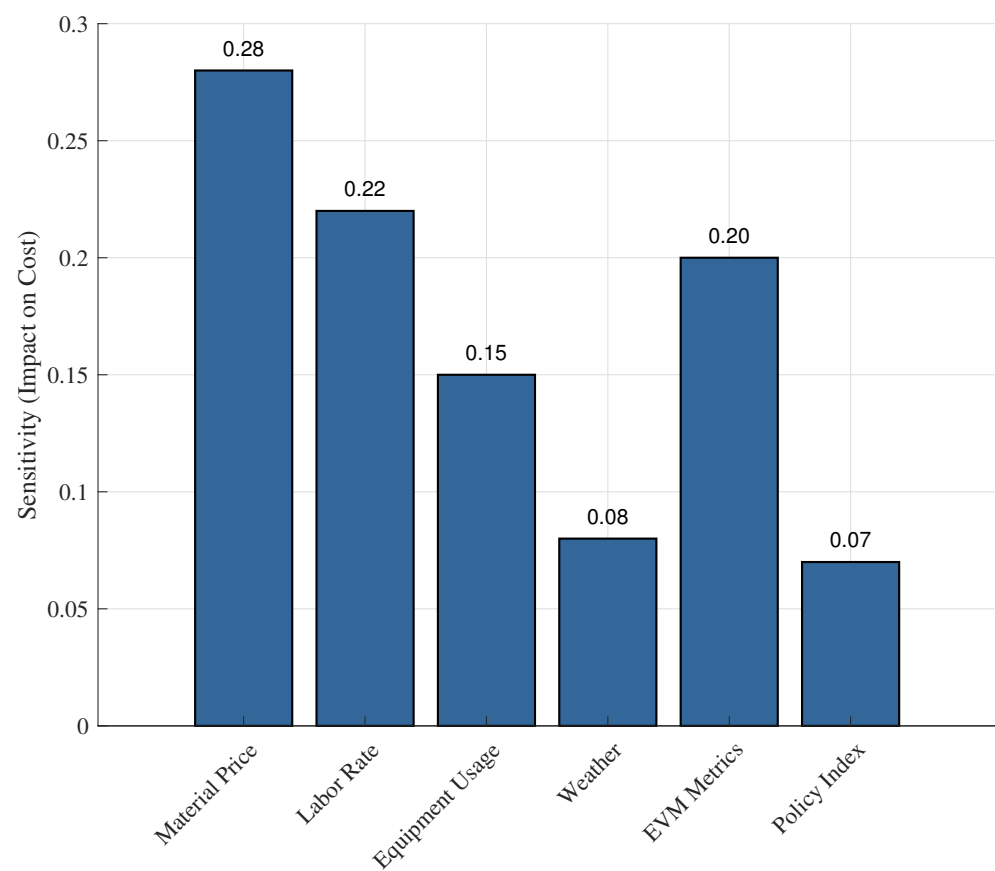


Figure 7. Feature importance ranking via sensitivity analysis. Material price, labor rate, and EVM metrics are the top three cost drivers.

4.3. Discussion of Model Superiority

To understand why the proposed model consistently outperforms the competing methods, we dissect the performance gains from four complementary perspectives.

First, the CP tensor fusion layer contributes substantially over naive concatenation. While ST-GCN employs graph convolution on flattened features, our CP decomposition explicitly preserves cross-source correlations, enabling the model to exploit shared information across heterogeneous data streams. Removing this fusion increases MAPE from 4.27% to 5.81%, confirming its critical role.

Second, the GCN module captures causal interrelations among cost drivers, e.g., steel price surges affect material costs and indirectly delay schedules, increasing labor and equipment expenses. This capability is absent in LSTM and Transformer, which treat each feature independently and rely solely on temporal patterns. The sensitivity analysis (Figure 7) validates this by identifying material price, labor rate, and EVM as the top three influences, consistent with the prior edges encoded in $\mathbf{A}_{\text{prior}}$.

Third, bidirectional temporal modeling and spatiotemporal attention enable the model to incorporate both past and future context within the lookback window. This is critical in construction cost forecasting, where cost variations often exhibit delayed responses. The attention heatmap (Figure 6) shows that weights shift toward material prices and EVM near the current time t , indicating adaptive focus on the most relevant information.

Finally, the composite loss function, particularly the temporal consistency term ($\lambda_2 = 0.05$), smooths unrealistic cost jumps common in purely MSE-optimized models. The ablation study confirms that removing this term increases MAPE to 5.02%, with more oscillatory predictions.

In summary, the superiority of the proposed model arises from the synergy of all four mechanisms—tensor fusion preserves data structure, GCN encodes driver interrelations, Bi-GRU captures temporal dynamics, and the composite loss ensures robust forecasts—rather than from any single component in isolation.

5. Conclusions and Future Work

This paper proposed a data-driven dynamic cost prediction framework for State Grid engineering projects, with its novelty rooted in the principled integration of tensor-based multi-source fusion, graph-structured driver modeling, and spatiotemporal attention within a unified architecture. Unlike existing approaches that apply generic forecasting methods or treat cost drivers independently, our design explicitly matches the intrinsic structure of the problem: CP decomposition preserves cross-source correlations, GCN captures causal interrelations, and Bi-GRU with attention handles delayed and time-varying responses. Extensive experiments on real substation projects demonstrate that this integration yields a 4.27% MAPE, reduces RMSE by 18.4% relative to LSTM, and maintains robust performance with up to 40% missing data. The attention heatmap and sensitivity analysis further provide interpretable insights, identifying material price and earned value metrics as the most influential factors—which is consistent with domain knowledge and validates the model's transparency.

Despite the promising results, several limitations should be acknowledged. First, model performance relies on data quality; severe missingness beyond the assumed 30% bound may impair imputation effectiveness. Second, the prior adjacency matrix is expert-derived and may not capture all latent dependencies, though the learnable residual partially mitigates this issue. Third, generalizability to projects with fundamentally different scales or regulatory contexts requires further validation. Fourth, the training cost and memory footprint may be substantial for large-scale deployment without hardware acceleration. These limitations inform our future research directions.

Future work will focus on online learning to adapt to concept drift over long-term project life cycles, transfer learning across multiple projects with limited labels, and integration with reinforcement learning for proactive cost control and resource reallocation. Additionally, extending the tensor fusion framework to handle streaming data and dynamic graph structures remains an important direction for real-time decision support in large-scale infrastructure construction.

Author Contributions: W.W. and Q.X. conceptualized the study and curated the data. T.Z. and F.F. acquired the funding and contributed to writing the original draft. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Young Scientists Fund, Grant No. 72401114) and the Key Research Base Projects of Humanities and Social Sciences in Jiangxi Provincial Universities (Project Number: JD23096).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data in this study cannot be provided publicly because they are protected by a signed confidentiality agreement. The related technical methods and procedures can be accessed according to the relevant confidentiality rules by contacting the author of this article agreement.

Conflicts of Interest: Authors Weiqiong Wang and Qidong Xu were employed by the company State Grid Jiangxi Electric Power Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AC	actual cost
ARIMA	autoregressive integrated moving average
Bi-GRU	bidirectional gated recurrent unit
CP	CANDECOMP/PARAFAC
CPI	cost performance index
EV	earned value
EVM	earned value management
GCN	graph convolutional network
GRU	gated recurrent unit
LSTM	long short-term memory
MAE	mean absolute error
MAPE	mean absolute percentage error
MNAR	missing not at random
MSE	mean squared error
PV	planned value
QS	quantile score
RMSE	root mean square error
SPI	schedule performance index
ST-GCN	spatiotemporal graph convolutional network
SVR	support vector regression
XGBoost	extreme gradient boosting

References

- Chen, L.; Lisha, Z.; Na, L.; Ming, Z. Modelling and simulation of power grid engineering project based on system dynamics on the background of smart grid. *Syst. Eng. Procedia* **2012**, *3*, 92–99. [\[CrossRef\]](#)
- Araiza, J.; Hambrick, J.; Moon, J.; Starke, M.; Vartanian, C. Grid energy-storage projects: Engineers building and using knowledge in emerging projects. *IEEE Electr. Mag.* **2018**, *6*, 14–19. [\[CrossRef\]](#)
- Liu, H. National quality infrastructure supports smart grid construction in china-taking the state grid as an example. In *Proceedings of the IOP Conference Series: Earth and Environmental Science*; IOP Publishing: Bristol, UK, 2020; Volume 531, p. 012011.
- Bradley, R. Pitfalls of using unit values as a price measure or price index. *J. Econ. Soc. Meas.* **2005**, *30*, 39–61. [\[CrossRef\]](#)
- Zhang, R.; Cheng, L.; Wang, S.; Lou, Y.; Gao, Y.; Wu, W.; Ng, D.W.K. Integrated sensing and communication with massive MIMO: A unified tensor approach for channel and target parameter estimation. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 8571–8587. [\[CrossRef\]](#)
- Jiang, S.; Hu, J.; Wood, K.L.; Luo, J. Data-driven design-by-analogy: State-of-the-art and future directions. *J. Mech. Des.* **2022**, *144*, 020801.
- Esi, M.; Wahyudi, A. Calculation of Production Prices Using the Full Costing Method and Cost of Sale Using the Cost Plus Pricing Method on D'atabajo Coffee. *ES Account. Financ.* **2024**, *2*, 130–137. [\[CrossRef\]](#)
- Hu, J.; Huang, H.; Chen, W.; Zhang, S.; Xia, D.; Yao, S. Research on optimization of power grid load forecasting models based on deep learning. *Sci. Rep.* **2026**. [\[CrossRef\]](#) [\[PubMed\]](#)
- Mathumitha, R.; Rathika, P.; Manimala, K. Intelligent deep learning techniques for energy consumption forecasting in smart buildings: A review. *Artif. Intell. Rev.* **2024**, *57*, 1. [\[CrossRef\]](#)
- Toosi, H.A.; Del Pero, C.; Leonforte, F.; Lavagna, M.; Aste, N. Machine learning for performance prediction in smart buildings: Photovoltaic self-consumption and life cycle cost optimization. *Appl. Energy* **2023**, *334*, 120648. [\[CrossRef\]](#)
- Tian, S.; Liu, X. Incorporating advanced machine learning algorithms into solar power forecasting in off-grid hybrid renewable systems. *Electr. Power Syst. Res.* **2025**, *248*, 111979. [\[CrossRef\]](#)
- Albeladi, K.; Zafar, B.; Mueen, A. Time series forecasting using LSTM and ARIMA. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 313–320. [\[CrossRef\]](#)
- Aziza, J.N.A. Perbandingan Metode Moving Average, Single Exponential Smoothing, dan Double Exponential Smoothing Pada Peramalan Permintaan Tabung Gas LPG PT Petrogas Prima Services. *J. Teknol. Manaj. Ind. Terap.* **2022**, *1*, 35–41. [\[CrossRef\]](#)
- Akomodi, J.O. The efficacy of statistics in all major fields of research: A focus on regression analysis. *Open J. Stat.* **2025**, *15*, 53–72. [\[CrossRef\]](#)
- Kontopoulou, V.I.; Panagopoulos, A.D.; Kakkos, I.; Matsopoulos, G.K. A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks. *Future Internet* **2023**, *15*, 255. [\[CrossRef\]](#)
- Ray, S.; Lama, A.; Mishra, P.; Biswas, T.; Das, S.S.; Gurung, B. An ARIMA-LSTM model for predicting volatile agricultural price series with random forest technique. *Appl. Soft Comput.* **2023**, *149*, 110939. [\[CrossRef\]](#)
- Sirisha, U.M.; Belavagi, M.C.; Attigeri, G. Profit prediction using ARIMA, SARIMA and LSTM models in time series forecasting: A comparison. *IEEE Access* **2022**, *10*, 124715–124727. [\[CrossRef\]](#)
- Liu, J. Navigating the financial landscape: The power and limitations of the ARIMA model. *Highlights Sci. Eng. Technol.* **2024**, *88*, 747–752. [\[CrossRef\]](#)
- Kavitha, S.; Varuna, S.; Ramya, R. A comparative analysis on linear regression and support vector regression. In *Proceedings of the 2016 Online International Conference on Green Engineering and Technologies (IC-GET)*; IEEE: New York, NY, USA, 2016; pp. 1–5.
- Rhodes, J.S.; Cutler, A.; Moon, K.R. Geometry-and accuracy-preserving random forest proximities. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 10947–10959. [\[CrossRef\]](#) [\[PubMed\]](#)
- Sibindi, R.; Mwangi, R.W.; Waititu, A.G. A boosting ensemble learning based hybrid light gradient boosting machine and extreme gradient boosting model for predicting house prices. *Eng. Rep.* **2023**, *5*, e12599.
- Luo, H.; Zhou, P.; Cui, J.; Wang, Y.; Zheng, H.; Wang, Y. Energy performance prediction of centrifugal pumps based on adaptive support vector regression. *Eng. Appl. Artif. Intell.* **2025**, *145*, 110247. [\[CrossRef\]](#)
- Douiba, M.; Benkirane, S.; Guezaz, A.; Azrour, M. An improved anomaly detection model for IoT security using decision tree and gradient boosting. *J. Supercomput.* **2023**, *79*, 3392–3411.
- Fekri, M.N.; Patel, H.; Grolinger, K.; Sharma, V. Deep learning for load forecasting with smart meter data: Online Adaptive Recurrent Neural Network. *Appl. Energy* **2021**, *282*, 116177. [\[CrossRef\]](#)
- Guo, Q.; He, Z.; Wang, Z. Assessing the effectiveness of long short-term memory and artificial neural network in predicting daily ozone concentrations in Liaocheng City. *Sci. Rep.* **2025**, *15*, 6798. [\[CrossRef\]](#) [\[PubMed\]](#)
- Niu, Z.; Zhong, G.; Yue, G.; Wang, L.N.; Yu, H.; Ling, X.; Dong, J. Recurrent attention unit: A new gated recurrent unit for long-term memory of important parts in sequential data. *Neurocomputing* **2023**, *517*, 1–9. [\[CrossRef\]](#)
- Altinses, D.; Löppenber, M.; Schwung, A. Efficient Heat Generation Systems Through Transformer and Recurrent Neural Forecasting Combined With Mixed-Integer Predictive Control. *IEEE Trans. Control Syst. Technol.* **2026**, *34*, 1259–1271. [\[CrossRef\]](#)

28. Moraes, C.P.; Fantinato, D.G.; Neves, A. Epanechnikov kernel for PDF estimation applied to equalization and blind source separation. *Signal Process.* **2021**, *189*, 108251. [[CrossRef](#)]
29. Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; Li, Y. Simple and deep graph convolutional networks. In *Proceedings of the International Conference on Machine Learning*; PMLR: London, UK, 2020; pp. 1725–1735.
30. Li, X.; Ma, X.; Xiao, F.; Xiao, C.; Wang, F.; Zhang, S. Time-series production forecasting method based on the integration of Bidirectional Gated Recurrent Unit (Bi-GRU) network and Sparrow Search Algorithm (SSA). *J. Pet. Sci. Eng.* **2022**, *208*, 109309. [[CrossRef](#)]
31. Cartis, C.; Gould, N.I.; Toint, P.L. Universal regularization methods: Varying the power, the smoothness and the accuracy. *SIAM J. Optim.* **2019**, *29*, 595–615. [[CrossRef](#)]
32. Seuring, S.A. Assessing the rigor of case study research in supply chain management. *Supply Chain Manag. Int. J.* **2008**, *13*, 128–137. [[CrossRef](#)]
33. Lewis, S. Qualitative inquiry and research design: Choosing among five approaches. *Health Promot. Pract.* **2015**, *16*, 473–475. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.