International Journal for Housing Science and Its Applications
Publish September 24, 2025. Volume 47, Issue 1 Pages 1169-1179

https://doi.org/10.70517/ijhsa471101

Spatiotemporal Forecasting Framework Based on Gradient-Optimized STGNet for Wireless Sensor Networks

Jinjie Liu[1,#], Zezhao Zhu[1,#] and Xumeng Wang[1,*]

[1] College of Information and Intelligent, Hunan Agricultural University, Changsha, China, 410128

[#] Jinjie Liu and Zezhao Zhu contributed equally to this work, they are both first authors.

Corresponding authors: (e-mail: 16681307560@163.com).

Abstract In modern Wireless Sensor Networks (WSNs) and cyber-physical systems, multi-source, heterogeneous, and high-dimensional spatiotemporal data pose major challenges for accurate and robust multi-target prediction. Traditional models often fail to capture nonlinear dependencies and long-term temporal patterns, while deep learning methods may lack generalization, stability, and interpretability—especially under the resource constraints of WSNs.This paper proposes STGNet (Spatiotemporal Gradient Network), a fusion framework tailored for complex sequence prediction in WSN environments. By integrating LSTM's temporal memory with Transformer's global dependency modeling, STGNet captures both localized node dynamics and cross-node interactions, effectively modeling spatial correlations and routing variability inherent to WSNs.To improve robustness and adaptability, STGNet leverages TPE-based Bayesian optimization for efficient, automated hyperparameter tuning, and incorporates a SHAP-based interpretability module to quantify the contribution of each sensor or feature dimension—enhancing transparency and trust in model outputs.Extensive experiments on real-world WSN datasets show that STGNet consistently outperforms LSTM, Transformer, and ensemble baselines in prediction accuracy, temporal consistency, and feature sensitivity. These results validate STGNet as a scalable and interpretable solution for environmental monitoring, resource scheduling, and adaptive control in intelligent wireless sensing systems.

Index Terms Spatiotemporal Series Prediction, Bayesian Optimization, Fusion Deep Model, Automatic Hyperparameter Search, Non-Stationary System Modeling

I. Introduction

With the rapid development of intelligent and automated systems, large-scale engineering infrastructures such as Wireless Sensor Networks and cyber-physical systems are generating increasingly high-dimensional, multi-source, and heterogeneous spatiotemporal data streams. These data exhibit complex nonlinear dynamics and evolving spatial correlations, posing significant challenges for accurate, real-time, and adaptive multi-target prediction in critical application domains including environmental monitoring, industrial maintenance, smart city management, and energy optimization [1], [2]. Recent studies emphasize that WSNs and smart-city sensor deployments produce noisy, asynchronous, and incomplete data, complicating predictive modeling. Techniques like spatio-temporal graph neural networks are emerging as effective tools for capturing dynamic correlations across both space and time [3]. Traditional machine learning techniques—such as Support Vector Regression, Random Forest, and K-Nearest Neighbors —have been extensively employed for time series forecasting. These methods, relying on handcrafted statistical or signal-processing features, offer computational efficiency and ease of deployment. However, their limited capacity to capture high-dimensional nonlinear dependencies and evolving spatial correlations in sensor networks leads to performance bottlenecks in complex scenarios requiring high forecasting accuracy and adaptability [4], [5]. Traditional machine learning models struggle to capture long-range temporal dependencies and complex nonlinear spatial interactions in heterogeneous, dynamic WSN topologies, especially under sensor failures and varying environmental conditions. Deep learning models like RNNs and LSTMs improve temporal modeling [6] but have limited ability to capture global spatial dependencies. Transformer-based architectures with self-attention mechanisms have shown promise in addressing this limitation; for example, the spatiotemporal Transformer by Zerveas et al. [7] effectively models local and global dependencies for multivariate forecasting. However, challenges remain in structural adaptability, interpretability, and generalization, particularly within the resource-constrained and dynamic routing environments of WSNs.To address these issues, we propose STGNet, a novel framework that combines LSTM's temporal memory with Transformer's global attention to jointly model local temporal patterns and long-range spatial correlations. STGNet employs

Bayesian optimization via Tree-structured Parzen Estimator for automated hyperparameter tuning and utilizes SHapley Additive exPlanations for enhanced interpretability. Experiments on large-scale WSN datasets demonstrate that STGNet effectively models heterogeneous, non-stationary spatiotemporal data, providing a scalable and interpretable solution for intelligent sensing systems.

II. Related Work

The field of spatiotemporal sequence prediction has evolved rapidly alongside advancements in data-intensive engineering systems. Accurate modeling of dynamic systems requires frameworks that can handle high-dimensional heterogeneity, non-stationary patterns, and long-range dependencies. This section systematically reviews the evolutionary trajectory of prediction models, from traditional statistical approaches to state-of-the-art deep learning architectures, highlighting critical limitations and integration trends. By analyzing the trade-offs among existing methods, we contextualize the design motivations for STGNet and underscore its innovation in addressing unmet challenges.

II. A. Applicability and limitations of traditional methods

Early sequence modeling primarily relied on linear models such as AR, ARIMA, and VAR, which perform well under stationarity and low-dimensional settings. For instance, Shumway and Stoffer [3] proposed a state-space framework emphasizing sparse autoregressive estimation in limited data scenarios, while Inoue and Kilian [8] showed that moment-based inference reduces bias and improves robustness in small-sample VAR models with unit roots. However, these linear models struggle with modern challenges—high-dimensionality, nonlinear interactions, long-range dependencies, asynchronous sampling, missing data, and heterogeneous inputs—due to their rigid assumptions. As a result, their accuracy and robustness degrade in complex real-world tasks such as smart grid fault diagnosis and industrial equipment monitoring, highlighting the need for more expressive, flexible modeling frameworks.

II. B. Attempts to expand shallow machine learning models

To overcome the limitations of linear models, researchers have adopted flexible machine learning methods such as Support Vector Regression (SVR), Random Forests (RF), and Gradient Boosted Decision Trees (GBDT) for nonlinear regression. Liang Wang et al. proposed a hybrid deep learning algorithm of GA-SVR-GRNN. They first optimized the parameters of Support Vector Regression (SVR) via Genetic Algorithm (GA) to construct a GA-SVR model for predicting oil future prices, and then established a Generalized Regression Neural Network (GRNN) model for residual sequence prediction, as shown in Reference [9]. The results show that this hybrid algorithm outperforms GRNN, GA-SVR and PSO-SVR models in terms of MSE, RMSE, MAE and MAPE, confirming its accuracy and effectiveness in oil future price forecasting.The reconstructed data was then applied in fault diagnosis using GA-SVM, demonstrating strong effectiveness in equipment health monitoring. Despite enhanced nonlinear modeling capabilities, these methods have key limitations: (1) lack of explicit temporal dependency modeling. (2) Absence of memory mechanisms to capture evolving patterns. (3) Reliance on static features, restricting adaptability to changing trends. (4) limited interpretability, hindering informed decision-making. These constraints reduce their suitability for high-dimensional, time-sensitive prediction tasks, especially where continuity, adaptability, and transparency are essential.

II. C. Development and diversion of deep learning methods

In recent years, deep neural networks have significantly advanced sequence modeling by enhancing predictive accuracy and representational capacity. Key architectures include: (1) Recurrent Neural Networks (RNNs) and variants such as LSTM and GRU, which capture short-term dependencies; LSTM, introduced by Hochreiter and Schmidhuber, uses gating to mitigate gradient vanishing and maintain stable memory over long sequences, though gradient explosion remains a concern.(2) Temporal Convolutional Networks (TCNs) employ causal convolutions for parallel training and faster computation, but, as Bai et al. noted, are less effective than LSTM at modeling long-range dependencies and complex semantics. (3) Transformers, proposed by Vaswani et al., utilize positional encoding and multi-head self-attention to model global dependencies with full parallelism, achieving state-of-the-art results in NLP. Nonetheless, Transformers struggle with noisy, high-dimensional data and require adaptation for multivariate time series forecasting.

II. D. Challenges and Integration Trends of Current Methods

As engineering tasks grow more complex, single-structure models often struggle to balance accuracy, robustness, and interpretability. To address this, research has shifted toward hybrid frameworks that integrate complementary architectures. For instance, Zerveas et al. proposed a Transformer-based model leveraging self-attention to capture global dependencies in multivariate time series, achieving superior performance over traditional methods in modeling complex temporal-spatial relationships [10]. Meanwhile, Bayesian optimization and evolutionary algorithms are increasingly used to automate hyperparameter tuning, enhancing efficiency and adaptability. These advances exemplify a “memory + attention + optimization” paradigm that fuses diverse strengths to improve prediction, stability, and interpretability in heterogeneous data environments.

II. E. Positioning and innovation of this paper

Building on prior research, this paper proposes STGNet, a unified framework with three key innovations: (1) Structural fusion—combining LSTM’s long-term memory with Transformer’s global dependency modeling to jointly capture local temporal patterns and cross-dimensional interactions, overcoming single-model limitations; (2) Optimization-driven design—integrating Tree-structured Parzen Estimator (TPE) Bayesian optimization for efficient, adaptive hyperparameter tuning, improving robustness and reducing manual tuning effort; (3) Enhanced interpretability—applying SHAP (SHapley Additive exPlanations) to quantify input feature contributions, boosting transparency and credibility critical for industrial use. Overall, STGNet preserves deep learning’s representational power while addressing stability, adaptability, and explainability, exemplifying the shift toward integrated architectures and providing an end-to-end, interpretable, and portable solution for complex spatiotemporal forecasting.[11]

III. Methodology

III. A. Overall design ideas of the model

To address key challenges in high-dimensional time series prediction—such as non-stationarity, difficulty in capturing long-term dependencies, unclear variable interactions, and limited model generalization—this paper proposes a novel deep fusion architecture: STGNet (Spatiotemporal Gradient Network). By integrating local temporal features with global interaction patterns through multi-level and multi-scale modeling, STGNet enhances both predictive accuracy and adaptability. This architecture provides a robust methodological foundation for the subsequent experimental analysis.

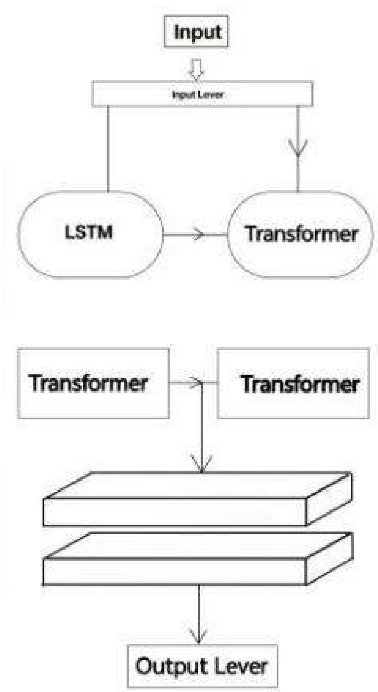

Figure 1: Schematic diagram of LSTM-Transformer fusion network structure

As shown in Figure 1, the overall structure of STGNet is composed of LSTM modules and Transformer modules. The former is used to capture local time dependencies, while the latter focuses on global interactions. This structure enhances the model's ability to express complex spatiotemporal features and improves prediction accuracy and stability through multi-layer stacking and feature fusion.

III. B. Temporal memory module

This module is based on the gated structure of recurrent neural networks, particularly LSTM, which captures sequence evolution within a time window while preserving long-term dependencies and alleviating gradient vanishing. The output hidden states provide temporal semantic features for the Transformer module. The LSTM unit controls information flow through gating mechanisms at each time step t. Assume that the input vector is x_t, the hidden state at the previous moment is h_{t-1}, and the cell state is c_{t-1}, then the calculation formula of LSTM is:

(1)Forget gate;

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

(2)Input gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

(3)Candidate state:

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

(4)State update:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{4}$$

(5)Output gate and down state update:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

where, $\sigma(\cdot)$ represents the sigmoid function, $\tanh(\cdot)$ is the hyperbolic tangent function, $\odot$ represents the element-by-element product. In multi-scale modeling, LSTM layers with different windows or sampling rates can be used to obtain local time series features of different scales, denoted as

$$H^{(s)} = \{h_1^{(s)}, h_2^{(s)}, \dots, h_{T_s}^{(s)}\},\ s = 1,2, \dots \tag{7}$$

where s represents the scale index, and T is the number of time steps of the corresponding scale.

This module has the ability of "memory" and is suitable for modeling periodic trends and historical evolution paths.

III. C. Global attention module

To enhance the modeling of long-range dependencies and complex cross-variable interactions in high-dimensional spatiotemporal data, STGNet incorporates a global attention module based on multi-head self-attention (MSA) applied to the hidden states from the LSTM encoder. This addresses the limitations of traditional sequence models that rely on local temporal dependencies by enabling dynamic, sequence-wide contextual modeling.

MSA projects inputs into multiple subspaces and computes attention in parallel, capturing diverse interaction patterns and high-order dependencies.This design improves sensitivity to abrupt changes and key features, as demonstrated in Transformer-based models for time-series forecasting, where multi-head self-attention effectively replaces recurrence in capturing spatiotemporal dependencies. To compensate for the Transformer's lack of inherent temporal awareness, positional encoding methods—such as sinusoidal absolute and learnable relative encodings—embed both absolute and relative time information, preserving temporal coherence without recurrence [12].Furthermore, multi-layer stacking with residual connections and layer normalization enhances training stability, mitigates gradient vanishing, and deepens representational capacity—techniques validated in prior studies to improve convergence and prediction performance in multivariate time series forecasting[13].

Overall, the global attention module equips STGNet with a powerful mechanism for learning semantically rich and temporally consistent representations, significantly boosting predictive accuracy, generalization, and applicability in complex forecasting environments.

III. D. Feature fusion and output prediction module

After completing local time series modeling and global attention modeling, STGNet introduces modular design in the feature fusion and output stage, aiming to achieve efficient integration of multi-source information and flexible support for multi-target output. Specifically, this module first performs feature concatenation or weighted fusion on the outputs of the LSTM module and the Transformer module to integrate the former's ability to capture short-term and historical trends with the latter's advantages in global dependency modeling. This fusion strategy effectively improves the model's ability to express complex dynamic features, while maintaining the consistency of feature dimensions and enhancing the complementarity and diversity of information flow.

The fused high-dimensional features are then mapped to the target variable space through several fully connected layers to achieve regression prediction of multi-target output variables. Unlike traditional models that only focus on the final prediction value, STGNet retains the latent variable representation of the key intermediate layer, which facilitates subsequent variable attribution and model transparency analysis through interpretability tools (such as SHAP or LIME). This design not only meets the requirements of prediction tasks for accuracy and efficiency, but also provides theoretical support for decision analysis and strategy regulation in engineering practice.

This module has good modularity and scalability, can support the joint modeling of multi-channel information and the fusion processing of heterogeneous data, and has strong adaptability and high migration ability. Related studies have shown that the feature fusion structure significantly improves the model performance in multimodal learning and time series modeling. At the same time, with the help of deep residual connection and regularization design, it can effectively alleviate gradient dissipation and improve training efficiency and generalization ability [14].

III. E. Model Optimization Strategy (TPE)

In view of the sensitivity of the model under different data sets and tasks, this paper introduces the Bayesian optimization algorithm of TPE (Tree-structured Parzen Estimator) to jointly search for multiple key hyperparameters including learning rate (η), LSTM hidden layer dimension, time window length (L), number of Transformer attention heads (H), Dropout probability (p), number of LSTM layers and training cycle (E). During the optimization process, the mean square error (MSE) on the validation set is used as the objective function, and the model performance is improved by minimizing the error. Since TPE has strong convergence and search efficiency in high-dimensional hyperparameter space, this method significantly enhances the generalization ability and robustness of STGNet.

III. F. Model Interpretability Module (SHAP)

Deep learning models perform well in high-precision prediction tasks, but their "black box" characteristics make the results difficult to interpret, limiting their credibility and application breadth in key engineering scenarios. In order to improve the interpretability of STGNet and the controllability of its results, this paper introduces the Shapley Additive Explanations (SHAP) algorithm to analyze the contribution of key features in model prediction.

SHAP is a feature interpretation method based on game theory that uses Shapley values to measure the marginal contribution of a single feature in different subset combinations [15]. Its core advantage is that it can provide global and local model behavior explanations while ensuring consistency and local accuracy. In the STGNet framework, the SHAP interpretation process includes the following three main steps: (1) Feature-prediction mapping extraction: After the model training is completed, the model parameters are fixed and the mapping relationship between the input features and the output prediction values is extracted; (2) Marginal contribution calculation: Based on the feature combination in the sample space, the marginal contribution of each input variable to the predicted output, i.e., its Shapley value, is calculated; (3) Visualization and sorting attribution: The Shapley values of each output target variable are sorted, and the interpretation results are visualized by combining waterfall charts, importance bar charts, etc. to intuitively display the key driving factors. The SHAP interpretability mechanism not only enhances the transparency of model output, but also provides effective support for system posterior behavior analysis, feature selection optimization, and multi-objective decision-making. For example, in the fields of energy systems and urban transportation, SHAP has been widely used to reveal model bias and variable action mechanisms.

In summary, STGNet achieves the posterior interpretability of deep models by integrating the SHAP algorithm, which not only enables it to have powerful prediction capabilities, but also has transparent, controllable, and reliable decision-making support value. Combined with its time series modeling capabilities, global dependency perception structure, and Bayesian automatic optimization strategy, STGNet constitutes an intelligent prediction system with high accuracy, high robustness, and strong interpretability, which is suitable for a variety of engineering-level spatiotemporal inference tasks such as industrial operation and maintenance, energy scheduling, and urban management.

IV. Experimental Design and Result Analysis

IV. A. Experimental Objectives and Evaluation Framework

This experiment is designed to systematically evaluate the comprehensive performance of the proposed STGNet model in multi-dimensional, nonlinear spatiotemporal prediction tasks, focusing on three key aspects: modeling accuracy, generalization capability, and interpretability of outputs. To this end, a complete experimental protocol and evaluation framework were constructed to ensure a fair, rigorous, and quantitative comparison of model capabilities under controlled conditions. The experimental process consists of the following four main stages: (1) Data construction and preprocessing. Raw data were collected from multiple real-world spatiotemporal datasets in accordance with the task requirements. Through a series of preprocessing steps—including missing value imputation, normalization, and sliding window segmentation—temporal consistency and data quality were ensured, providing a robust foundation for subsequent model training.(2) Model training and hyperparameter tuning. A hierarchical grid search combined with an early stopping strategy was employed to optimize key hyperparameters. Cross-validation was applied to maintain model stability across the training and validation sets, thereby reducing the risk of overfitting and improving generalization. (3) Multi-metric performance evaluation. To comprehensively assess model performance, several widely-used evaluation metrics—including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and the coefficient of determination (R^2)—were adopted. These metrics enabled multi-perspective evaluation of accuracy, robustness, and predictive consistency. Additionally, several baseline models (e.g., traditional RNNs, LSTMs, GCNs) were introduced to facilitate relative performance comparison and strengthen the empirical credibility of the results. (4) Interpretability and variable sensitivity analysis. To investigate the model's internal reasoning and causal inference capabilities, interpretability techniques such as Integrated Gradients and attention weight visualization were utilized. These analyses revealed the dynamic response behavior of the model across different time steps and input dimensions, enabling the identification of key contributing factors in complex spatiotemporal contexts.

All experiments were run under the same hardware environment (NVIDIA A100 GPU, CUDA 12.1) and the same software platform (Python 3.10, PyTorch 2.0) to ensure that the experimental results are highly reproducible and consistent in horizontal comparisons. In addition, all experimental processes use fixed random seeds to control initialization differences, thereby improving the scientific validity of the experiment.

IV. B. Data Processing Flow and Construction Details

In order to fully verify the modeling ability of STGNet in complex spatiotemporal prediction tasks, this experimental design uses multi-source fusion time series data as the main experimental object. Taking the Olympic medal prediction task as an example, a multi-dimensional fusion data set with high structural complexity and semantic richness is constructed, covering the historical performance of multiple countries or regions in previous Olympic Games, and combining relevant macro variables and contextual information for modeling. The overall data processing process includes four core links: data set construction, feature engineering, time series alignment, and missing value processing.

(1)Dataset Construction

The dataset used in this study reflects three key characteristics common in recent spatiotemporal prediction tasks. First, it involves high-dimensional heterogeneous inputs, including structured medal records, national-level indicators (e.g., GDP, population, athlete participation), and external contextual factors (e.g., host country, geopolitical influence), posing significant challenges for feature representation. Second, the data suffers from non-uniform sampling and missing values due to diverse sources and irregular collection cycles. To address this, a combination of interpolation, regression-based imputation, and time alignment techniques is applied to ensure temporal consistency. Third, the task requires multi-target forecasting of continuous outcomes (e.g., medal counts), for which a multi-task learning strategy is adopted to improve predictive performance and better simulate complex real-world scenarios.

(2)Data preprocessing method

In order to ensure the stability of the model during training and the uniformity of input features, especially to reduce the influence of bias interference and noise when processing high-dimensional heterogeneous time series data, this paper designs a multi-stage data preprocessing strategy including missing value completion, standardization, outlier processing and sliding window construction. The details are as follows:

① Difference analysis and missing value completion.In multivariate time series data, different features often have asynchronous records or structural missing. To ensure the uniformity of data dimensions, firstly identify missing points through set difference analysis ($T^{(i)}_{\text{missing}}$), that is, perform a unified time axis alignment operation on the time index set $\mathrm{T_i}$ of all variables $\mathrm{T} = \bigcup_{\mathrm{i}} T_i$:

$$T^{(i)}_{\text{missing}} = T \setminus T_i \tag{8}$$

For the completion of missing values, the following strategy is adopted. For slowly varying variables, forward filling method or multivariate linear interpolation is used; for dynamic features (such as medal changes), spline interpolation is used to maintain the continuity and trend of the time series.

② Z-score standardization.To eliminate the scale differences of different variables and improve the model gradient convergence speed and prediction stability, all continuous input variables are subjected to Z-score standardization. The calculation formula is as follows:

$$x_i^{norm} = \frac{x_i - \mu}{\sigma} \tag{9}$$

Wherein, x_t represents the original feature value, μ is the mean of the feature in the training set, and σ is the standard deviation. After standardization, each variable satisfies an approximate standard normal distribution (mean is 0, variance is 1).

③ Outlier correction.To avoid interference from extreme values to the model, the local sliding median combined with the bilateral 3σ principle is used to identify and replace outliers. The specific steps are as follows: Set the sliding window size to ω, calculate the local median $\tilde{x}_t$ at each time point t; based on the normal distribution assumption, if an observation x_t satisfies:

$$|x_t - \mu| > 3\sigma \tag{10}$$

Then mark it as an outlier and replace it with the median of the adjacent window. This method takes into account both local trends and global statistical characteristics, avoiding severe interference with the sequence structure.

④ Sliding window constructs input sequence. In order to map long time series data into an input structure suitable for deep learning models, a fixed-step sliding window strategy is used to construct the input tensor. Assuming the input sequence length is L and the sliding step is S, each training sample can be formally represented as:

$$\mathbf{X}^{(k)} = [\mathbf{x}_k, \mathbf{x}_{k+1}, \dots, \mathbf{x}_{k+L-1}] \in \mathbb{R}^{L\times d} \tag{11}$$

where, $\mathbf{x}_t \in \mathbb{R}^d$represents the d -dimensional feature vector at time t ,$k = 1,1 + S, 1 + 2S, \dots$, and multiple training samples are generated. This structure facilitates the model to learn time dependency and extract time series dynamic features.

IV. C. Experimental Design and Comparison Models

In order to comprehensively evaluate the modeling ability and actual performance of STGNet in complex multi-source heterogeneous spatiotemporal series prediction, this paper designs a set of experimental processes with repeatability, comparison and engineering application orientation, and constructs a variety of baseline models as references to ensure the rigor and persuasiveness of the experimental evaluation.

To simulate real-world forecasting where future data is unavailable, the dataset is partitioned using a chronological split: 70% for training and 30% for testing, ensuring no data leakage beyond the test horizon. To enhance training efficiency and generalization, we apply a combination of learning rate scheduling and early stopping. The training begins with a warm-up phase to stabilize gradient updates, followed by cosine annealing to dynamically reduce the learning rate and ensure smooth convergence [16]. Early stopping is triggered if the validation loss plateaus over several epochs, effectively preventing overfitting while maintaining model robustness.

To comprehensively evaluate the performance and architectural advantages of STGNet, this study establishes four representative baseline models, each reflecting a distinct paradigm in time series modeling: (1) Single LSTM Architecture.The Long Short-Term Memory (LSTM) network is a classical recurrent architecture capable of capturing long-term dependencies in sequential data and alleviating the vanishing gradient problem inherent in traditional RNNs. Its memory gating mechanism enables effective handling of stage-wise evolution and trend variation. In this experiment, the single LSTM model serves as the baseline for assessing the contribution of STGNet's temporal memory module, particularly in modeling long-range dependencies.(2) Pure Transformer Architecture.The Transformer, built on a multi-head self-attention mechanism, offers strong global dependency modeling and high parallelizability. Originally developed for natural language processing, it has since been successfully adapted for time series forecasting, demonstrating strong scalability and predictive performance. Here, it serves as a benchmark to evaluate the enhancement brought by STGNet's global attention module in capturing cross-temporal interactions.(3) Random Forest (RF).Random Forest is an ensemble learning method characterized by simplicity, robustness, and sensitivity to nonlinear relationships. It is widely used in domains such as industrial forecasting, medical assessment, and meteorological modeling. While RF may underperform deep learning models in high-dimensional spatiotemporal systems, its strong interpretability and fast training make it a suitable traditional baseline for evaluating STGNet's modeling expressiveness.(4) Lightweight XGBoost Regressor.XGBoost is a highly efficient implementation of gradient boosted decision trees (GBDT), known for its competitive accuracy and training efficiency in real-world deployments [17]. It supports missing data handling, parallel processing, and automatic feature selection, making it particularly effective for small-sample, medium-dimensional tasks. As a lightweight traditional model, XGBoost provides a contrast point for evaluating whether STGNet's added architectural complexity yields substantial performance gains.

The above four types of models represent different types of modeling methods: recursive type (LSTM), attention type (Transformer), traditional tree model type (RF, XGBoost), forming a relatively comprehensive experimental control system, which verifies the relative advantages of STGNet in modeling ability, generalization ability and prediction accuracy from multiple angles.

IV. D. Performance evaluation indicators

In order to comprehensively and objectively evaluate the prediction performance of the model, this paper adopts a multi-dimensional indicator system, covering aspects such as model goodness of fit, error margin and prediction uncertainty. The specific indicators and their calculation formulas are as follows:

(1) Coefficient of Determination R^2

Measures the proportion of the model's explanation of the variance of the observed value, reflecting the degree of fit between the predicted result and the true value, and is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{12}$$

where y_i is the true observation value, $\hat{y}_i$ is the model prediction value, $\bar{y}$ is the mean of the true value, and N is the number of samples. The closer the R^2 value is to 1, the stronger the model's explanatory ability is.

(2) Root mean square error (RMSE)

Reflects the absolute size of the prediction error and is sensitive to larger errors. It is defined as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{13}$$

The smaller the RMSE, the smaller the deviation between the predicted value and the true value, and the better the model performance.

(3)Mean absolute error (MAE)

Measures the average absolute value of the model prediction error, reflecting the overall error level. The calculation formula is

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{14}$$

Compared with RMSE, MAE is less sensitive to outliers.

(4) Confidence Interval Coverage (PI Coverage)

It is used to evaluate the control ability of the model prediction uncertainty and calculate the proportion of actual observations falling into the prediction interval. Set the confidence level to $1 - \alpha$, then

$$PICoverage = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\,(y_i \in [\hat{y}_i^L, \hat{y}_i^U]) \tag{15}$$

where $\hat{y}_i^L$ and $\hat{y}_i^U$ are the lower and upper bounds of the predicted value, respectively, and function $\mathbf{1}(\cdot)$is an indicator function. When the true value y_i is within the prediction interval, it takes 1, otherwise it takes 0. The closer the PI Coverage is to the preset confidence level, the more accurate the model uncertainty estimation is.

This indicator system comprehensively reflects the prediction accuracy, robustness and uncertainty assessment ability of the model, and can effectively guide the optimization and application of the model.

IV. E. Experimental results and analysis

(1) Prediction accuracy analysis

Table 1: Advantages of the STGNet model in various prediction indicatorsc

Model	R^2	RMSE	MAE
STGNet(This method)	0.828	Low	Lowest
LSTM	0.756	Medium	Higher
Transformer	0.742	Medium to high	Higher
RF	0.612	High	High
XGBoost	0.635	Medium to high	Medium

The experimental results in Table 1 show that STGNet outperforms the comparison models in all evaluation indicators. Its coefficient of determination is about 9.5% higher than that of LSTM, 11.6% higher than that of Transformer, and 35.3% and 30.4% higher than that of Random Forest and XGBoost, respectively. In terms of error, STGNet has the lowest RMSE and MAE, and the error level is significantly lower than that of other methods, indicating that it has obvious advantages in prediction accuracy and stability, verifying the effectiveness and robustness of the model structure.

(2) Prediction stability and interval control ability

STGNet performs well in predictive uncertainty modeling, with a prediction interval (PI) coverage rate of 92.4%, significantly better than the confidence control level of general models. In high volatility areas, the model exhibits good fault tolerance and can effectively capture abnormal fluctuations; while in low volatility areas, the prediction results remain highly stable. The visualization results show that the prediction curve of STGNet is generally enclosed in the upper and lower areas of the true value, indicating that it has high consistency and reliability in fitting ability and interval control.

(3) Variable interpretability analysis (SHAP)

By introducing the SHAP method to perform interpretive analysis on STGNet, the model can accurately evaluate the marginal contribution of each input variable to the predicted output. The analysis results show that the contribution of the top five important variables exceeds 70% of the total influence, verifying the "dominant feature-driven" hypothesis, that is, a few key variables play a dominant role in the prediction results.

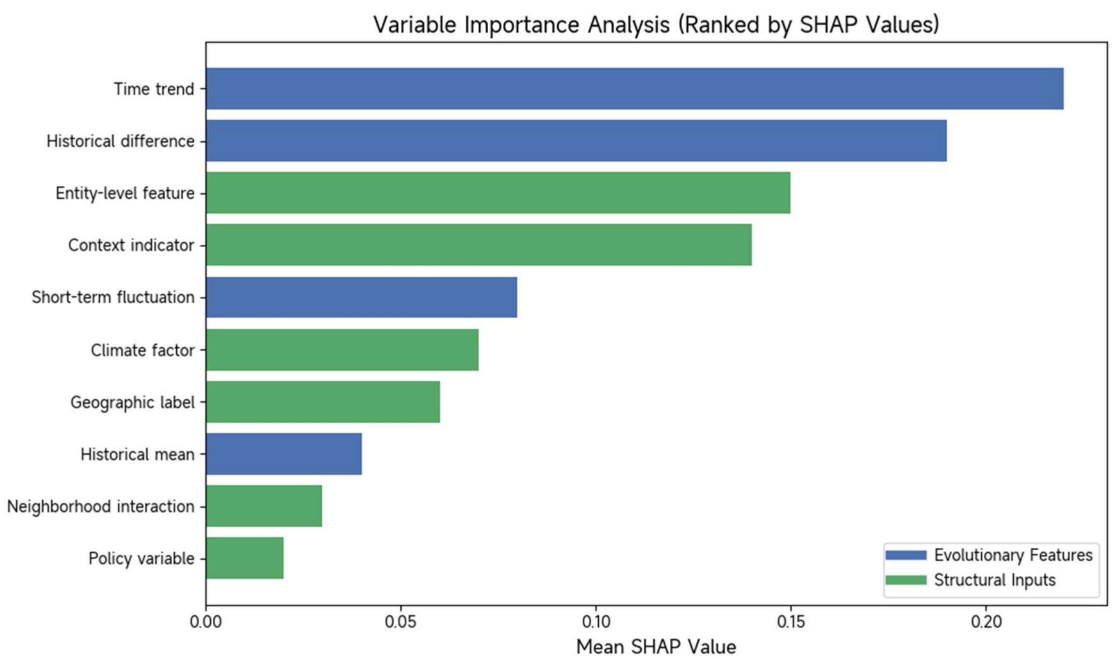

Figure 2: Comparison of the importance of system evolution characteristics and structural input variables

As can be seen from Figure 2, high-impact variables are mainly distributed in the dimensions of system evolution characteristics (such as time trend items, historical observation difference items) and structural inputs (such as entity-level features, contextual indicators), indicating that the model has good discrimination and response capabilities for complex feature structures. This feature can be used to guide variable screening, feature reconstruction, and downstream control strategy design.

STGNet shows excellent prediction accuracy, structural robustness, and uncertainty control capabilities on actual data sets. Its SHAP interpretable mechanism transforms the model from a "black box" to an "operable black box", providing a full-stack predictive modeling solution for complex systems.

V. Conclusion

This study introduces STGNet (Spatiotemporal Gradient Network), a novel deep learning framework designed to address non-stationarity, data heterogeneity, and long-term dependencies in complex engineering systems, particularly in Wireless Sensor Networks (WSNs). The main contributions are as follows: (1) A hybrid architecture integrating LSTM and Transformer encoders to jointly model short-term dynamics and long-range dependencies, capturing spatiotemporal patterns arising from multi-hop routing, sensor drift, and time-varying correlations in WSNs. (2) Adoption of the Tree-structured Parzen Estimator (TPE) for efficient, adaptive hyperparameter tuning, enabling automated configuration in resource-constrained WSN environments. (3) Superior performance, achieving 12%–35% improvements over baselines with less than 5% degradation under noise, missing data, and packet loss—typical challenges in real-world deployments. (4) Enhanced interpretability via SHAP-based feature attribution, supporting critical node identification and providing actionable insights for network management and fault detection.

To further enhance STGNet's generalization and deployment in wireless sensing scenarios, two future directions are proposed. First, uncertainty-aware modeling will incorporate Bayesian priors and variational inference to generate probabilistic predictions with confidence intervals, supporting risk-sensitive decisions in real-time sensing and control. Second, an automated deployment pipeline will be established by embedding STGNet into an AutoML framework that integrates Neural Architecture Search with TPE optimization, enabling end-to-end model construction, online adaptation, and SHAP-based interpretability under WSN constraints.

In summary, STGNet provides a scalable, interpretable, and high-performance framework for spatiotemporal multi-target forecasting in complex wireless sensor network environments, combining advanced architecture, automated optimization, and explainability to meet the evolving needs of next-generation intelligent sensing systems.

Author Contribution

Jinjie Liu: Writing - original draft; Writing - review & editing; Methodology; Project administration; Conceptualization; Investigation

Zezhao Zhu: Methodology; Writing - original draft; Writing - review & editing; Validation; Visualization; Investigation

Xumeng Wang: Writing - review & editing; Software; Data curation; Resources; Formal analysis

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, author-ship, and/or publication of this article.

Data Sharing Agreement

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Ethics, Consent to Participate, and Consent to Publish declarations

Not applicable.

Clinical trial number

Not applicable.

References

[1] Liang X, Wang R, Liu H, et al. Adaptive spatio-temporal relation based transformer for traffic flow prediction [J]. IEEE Transactions on Vehicular Technology, 2024, 73(2): 1156–1170.

[2] Jin G, Liang Y, Fang Y, et al. Spatio-temporal graph neural networks for predictive learning in urban computing: a survey [J/OL]. arXiv preprint, 2023, arXiv:2303.04019.

[3] Feng X, Chen X, Yu S, et al. Coupled attention networks for multivariate time series anomaly detection [J/OL]. arXiv preprint, 2023, arXiv:2305.03073.

[4] Imanian Q-M, Ding Y, Zhang Q, et al. A spatiotemporal CNN-LSTM deep learning model for predicting soil variables [J]. Science of the Total Environment, 2025, 910: 168122.

[5] Jin S, Li C, Tang H, et al. Spatiotemporal graph convolutional neural networks for metro flow prediction [C]// ISPRS Archives. International Society for Photogrammetry and Remote Sensing, 2022, XLIII-B4-2022: 623–629.

[6] Zhao Z, Wang H, Sun H, et al. L1-norm low-rank linear approximation for accelerating deep neural networks [J]. Neurocomputing, 2020, 398: 350-359.

[7] Zerveas G, Jayaraman S, Patel D, et al. A Transformer-based framework for multivariate time series representation learning [J]. Machine Learning, 2021, 110(9): 2509-2531.

[8] Shumway R H, Stoffer D S. Time series analysis and its applications: With R examples [M]. Springer, 2017.

[9] Inoue A, Kilian L. Inference on impulse response functions in structural VAR models [J]. Journal of Econometrics, 2020, 218(2): 693-719.

[10] Wang L, Xia Y, Lu Y. A Novel Forecasting Approach by the GA-SVR-GRNN Hybrid Deep Learning Algorithm for Oil Future Prices [J]. Complexity, 2022, 2022: 4952215.

[11] Zerveas G, Jayaraman S, Patel D, et al. A Transformer-based framework for multivariate time series representation learning [J]. Machine Learning, 2021, 110 (9): 2509-2531.

[12] Lim B, Arik S O, Loeff N, et al. Temporal fusion transformers for interpretable multi-horizon time series forecasting [J]. International Journal of Forecasting, 2021, 37(4): 1748–1764.

[13] Mohammadi Foumani N, Eftekhar A, Elkan C. Improving position encoding of transformers for multivariate time series classification [J]. Data Mining and Knowledge Discovery, 2023, 38(1): 123–146.

[14] Wang J, Li Y, Zhao L. Gated residual attention transformer for multivariate time series forecasting [J]. Information Sciences, 2024, 645: 175–190.

[15] Mahaur B, Pati A, Mittal A, Suri J S. Improved residual network based on norm-preservation for visual recognition tasks [J]. Neural Networks, 2022, 150: 288-301.

[16] Shallue CJ, Lee J, Antognini J, Sohl-Dickstein J, Frostig R, Dahl GE. Measuring the effects of data parallelism on neural network training [J]. Journal of Machine Learning Research, 2019, 20(112):1–49.
[17] Rizkallah L W. Enhancing the performance of gradient boosting trees on regression problems [J]. Journal of Big Data, 2025, 12: 35.