Journal of Computer Science and Artificial Intelligence
ISSN: 3078-8242 (Print), ISSN: 3078-2309 (Online) | Vol. 7, No. 1, 2026

A Hybrid Model for Multi-step Time Series Prediction Based on Granular Computing and LSTM

Zhitao Jia, Lijie Jia *, Lianchao Qiu

School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo 454000, China

* **Corresponding author**: (Email: jialijie@126.com)

Abstract: Multi-step time series forecasting based on iterative strategies often suffers from severe error accumulation, which significantly degrades long-horizon prediction performance. To address this issue, this paper proposes a hybrid forecasting framework that integrates granular computing with Long Short-Term Memory (LSTM) networks. First, the original time series is transformed into a sequence of temporal information granules, each represented by a triplet consisting of cardinality, mean, and standard deviation, which respectively characterize data density, trend level, and fluctuation intensity. Second, three p arallel LSTM networks are constructed to independently predict these granular compon ents, enabling effective feature decoupling and reducing mutual interference during the learning process. Finally, a degranulation mechanism is employed to reconstruct the predicted granules into the original time series, allowing multiple future points to be forecasted within a single prediction step and thereby mitigating error propagation. Extensive experiments on diverse public benchmarks, including classical time series datasets and multi-city PM2.5 concentration data, demonstrate that the proposed framework consistently outperforms traditional statistical methods, standard LSTM models, and existing granulation-based approaches, with im-proved accuracy under long-horizon fore-casting scenarios.

Keywords: Multi-step prediction, LSTM, granulometric computation, errors accumulation.

1. Introduction

Time series forecasting has become an essential analytical tool in many fields, including economics [1-3], environmental science [4-5], healthcare [6], and industrial systems [7]. By leveraging historical observations, forecasting models aim to capture the underlying evolution of system dynamics and provide reliable support for long-term planning and strategic decision-making. According to the length of the prediction horizon, forecasting tasks are typically categorized into single-step and multi-step scenarios. While single-step forecasting focuses on predicting the immediate next value, multi-step forecasting aims to predict multiple future points and is therefore more valuable for long-term analysis and control tasks.

Classical statistical models, such as the Auto-Regressive Moving Average (ARMA) model and its extensions, have long been adopted for multi-step time series forecasting [8-11]. These approaches typically generate multi-step predictions through iterative strategies, whereby the output at each step is recursively fed back as input for the subsequent pre-diction. Representative applications include ARX-based wind power forecasting [12], electricity load prediction using ARMA models [13], and joint forecasting of wind speed and direction [14]. Nevertheless, such models are fundamentally built upon restrictive assumptions, including stationarity and linear dependence [15-17]. In practical scenarios, real-world time series frequently exhibit pronounced nonlinearity, volatility, and intricate temporal dependencies, thereby constraining the predictive capability of conventional statistical frameworks.

Driven by advances in data-driven modeling, Artificial Neural Networks (ANNs), particularly Re-current Neural Networks (RNNs) [18] and Long Short-Term Memory (LSTM) networks [19], have emerged as prominent tools for time series forecasting. Benefiting from gating mechanisms and memory units, LSTM networks are capable of capturing long-term dependencies and modeling nonlinear temporal dynamics. Recent studies have further developed hybrid deep learning frameworks, including swarm intelligence–enhanced LSTM models [20, 21], decomposition-based approaches integrating CEEM-DAN with deep networks [22], and spatiotemporal CNN–LSTM architectures designed to characterize complex spatial–temporal interactions [23]. Despite these advances, LSTM-based models are often criticized for their limited interpretability, as they typically function as black-box predictors [24]. More critically, most LSTM-based multi-step forecasting strategies continue to adopt iterative prediction mechanisms, under which prediction errors are recursively propagated and gradually amplified as the forecasting horizon extends. Such cumulative error accumulation substantially undermines forecasting reliability in long-horizon scenarios and remains a fundamental challenge in multi-step time series prediction.

In parallel, representation learning and abstraction-oriented approaches have been widely explored to improve the robustness of multi-step time series forecasting. Decomposition and feature trans-formation methods extract trend, seasonal, and residual components prior to prediction, while fuzzy and probabilistic models emphasize uncertainty modeling and interpretability. Recently, granular computing has attracted increasing attention as an abstraction-driven framework that reduces forecasting complexity by operating on information granules rather than raw data. In this setting, time series are represented as sequences of granules, each summarizing a subsequence at a higher level of abstraction, thereby reducing iterative prediction steps and mitigating error propagation. Representative studies include clustering-based granular modeling [25], fuzzy cognitive maps for multi-level abstraction [26], linear fuzzy Gaussian granules incorporating temporal information [27], trend-based fuzzy granulation with LSTM [28], and triplet-based granulation

combined with probabilistic forecasting [29]. Despite these advances, many approaches rely on rule-based or fuzzy inference mechanisms, limiting their ability to capture complex nonlinear dependencies. In addition, certain granulation strategies introduce considerable computational overhead, which restricts their scalability in large-scale and long-horizon forecasting.

To address these limitations, a hybrid multistep forecasting framework integrating granular computing and LSTM networks is developed. The key idea is to reformulate point-wise forecasting into a granule-wise paradigm. The original time series is segmented into temporal granules, each represented by cardinality, mean, and standard deviation. Com-pared with the covariance-based representation in [29], this design provides a more efficient and interpretable description for univariate data while preserving essential fluctuation characteristics. Three parallel LSTM networks are then used to predict these components independently, enabling feature decoupling and reducing training interference. Finally, a degranulation step reconstructs the predicted granules into the original sequence, generating multiple future values in a single step and alleviating error accumulation in long-horizon forecasting.

The main contributions of this study are summarized as follows:

1) A hybrid multi-step time series forecasting framework integrating granular computing with LS-TM networks is developed, reformulating conventional point-wise prediction into a granule-wise paradigm to mitigate error accumulation.

2) Feature decoupling is realized through a parallel LSTM architecture that independently models the cardinality, mean, and standard deviation of each temporal granule, thereby improving predictive accuracy and enhancing training stability.

3) An efficient granulation strategy is devised to construct temporal information granules with reduced computational complexity while preserving essential statistical characteristics of the original sequenc

4) Extensive experiments conducted on multiple public benchmark datasets, including classical time series and large-scale $PM_{2.5}$ data, validate the effectiveness and robustness of the proposed framework. Furthermore, comprehensive granule-level error analyses substantiate its stability and reliability, particularly under long-horizon forecasting scenarios.

The remainder of this paper is organized as follows. Section II introduces the fundamental concepts of LSTM networks and granular computing. Section III details the proposed granulation-based LSTM forecasting framework. Section IV reports experimental results, comparative evaluations, and granule-level error analyses to assess the performance and robustness of the proposed approach. Finally, Section V concludes the paper and outlines directions for future research.

2. Prerequisites

This section introduces the fundamental concepts underpinning the proposed framework, with emphasis on LSTM networks and granular computing. The properties of LSTM networks are examined in relation to their gated architecture and capability to capture long-term dependencies, whereas granular computing is characterized as a multi-granularity abstraction paradigm that simplifies data through well-justified partitioning while preserving essential information.

2.1. LSTM Neural Network

Traditional RNNs can process sequential data but often suffer from vanishing and exploding gradients when handling long sequences, which limits their ability to capture long-range dependencies. LSTM networks address this issue by introducing a gated memory mechanism that regulates information retention and forgetting. The cell state enables stable information flow and supports effective gradient propagation over time, while the input, forget, and output gates selectively preserve relevant temporal features, allowing more reliable modeling of long-term dependencies.

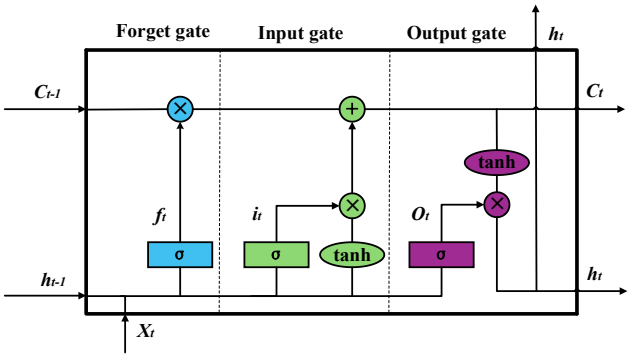

Figure 1. Operating Principle of LSTM

The core architecture of an LSTM network comprises a memory cell and a gating mechanism. The memory cell (C) serves as the primary component for preserving long-term information, while the gating mechanism dynamically modulates the cell state and governs information flow over time. As illustrated in Fig. 1, a standard LSTM unit incorporates three gates—namely, the input gate, forget gate, and output gate—denoted by i, f, and o, respectively. Among them, the forget gate determines which historical information is retained within the memory cell and is mathematically formulated as follows:

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \tag{1}$$

Where x_t denotes the input at time t, h_{t-1} represents the hidden state at time t-1, $\sigma(\cdot)$ denotes the sigmoid activation function, W_{fx} and W_{fh} are the weight matrices corresponding to the current input and the previous hidden state, respectively, and b_f is the bias term associated with the forget gate.

The input gate determines whether information from the current input should be incorporated into the memory cell and is mathematically defined as follows:

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \tag{2}$$

Where W_{ix} and W_{ih} denote the weight matrices of the input gate corresponding to the current input and the previous hidden state, respectively, and b_i is the bias term of the input gate.

The cell state is updated as follows:

$$c_t = tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \tag{3}$$

Where W_{cx} and W_{ch} are the weight matrices associated with the input and the previous hidden state, respectively, and b_c denotes the bias term of the candidate state.

The state of the memory cell at time t is expressed as follows:

$$C_t = f_t \circ C_{t-1} + i_t \circ c_t \quad (4)$$

The effect of the memory cell on the current output, controlled by the output gate, is defined as follows:

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \quad (5)$$

Where W_{ox} and W_{oh} are the weight matrices of the output gate associated with the current input and the previous hidden state, respectively, and b_o is the bias term of the output gate.

Finally, the hidden state is computed as follows:

$$h_t = o_t \circ tanh(C_t) \quad (6)$$

2.2. Granular Computing

Transforming the original time series into granular sequences enables deeper information abstraction and supports multi-dimensional analysis. Compared with raw temporal observations, granular representations provide a more compact structure and improved computational efficiency. Consequently, the segmentation strategy—that is, how to partition the series into meaningful information granules—has become a central issue in granular time series modeling.

The principle of justifiable granularity [30] ensures that each constructed granule is semantically meaningful and appropriately sized, thereby enabling a representative abstraction of the original data. De-spite these conceptual advantages, its practical implementation poses a fundamental challenge: achieving a balance between the competing objectives of specificity and coverage.

Coverage: Each information granule should be supported by a sufficient number of data samples to ensure statistical reliability. In particular, a qualified granule must exhibit adequate coverage of the underlying data. The coverage of a granule Ω_m is characterized by its cardinality n, defined as card$\{x_j|x_j \in \Omega_m\}$, where $m \in \{1, 2, \ldots, i\}$. In other words, the cardinality represents the number of data points contained within the granule.

Specificity: An effective information granule should strike a balance between adequate coverage and high specificity, where specificity reflects the granule's ability to capture distinctive internal characteristics. The standard deviation quantifies the dispersion of a temporal segment around its mean and serves as a primary indicator of intra-granular variability. Larger values indicate stronger volatility ar-ising from abrupt fluctuations or external disturbances, whereas smaller values imply greater stability and regularity, which are typically associated with higher predictability. By incorporating the standard deviation into the granulation process, essential dynamic features are preserved, thereby improving both fore-casting accuracy and model interpretability.

Remark 1. Although covariance matrices are commonly used to describe structural relationships in multivariate data, standard deviation provides a more efficient and interpretable alternative for univariate time series. The covariance matrix requires a computational complexity of $O(Nd^2)$, where N is the sequence length and d is the feature dimension, whereas standard deviation can be computed in linear time $O(N)$. Consequently, using standard deviation in the granulation process reduces computational over-head while preserving essential fluctuation characteristics, especially when a large number of granules are involved.

In accordance with the principle of justifiable granularity, a fundamental challenge lies in balancing two competing criteria. The quality of a granule is quantified by Equation (7):

$$Q(\Omega_m) = card\{x_j \mid x_j \in \Omega_m\} \cdot exp^{-\sqrt{s_m}} \quad (7)$$

Which is characterized by two counteracting components: the first term, modeled as a monotonically increasing function of cardinality, and the second term, defined as a decreasing function of the standard deviation. The interplay between these opposing terms serves as a mechanism to guide the formation of granules in a statistically justifiable and par-simonious manner.

In addition to standard deviation and cardinality, the mean is introduced as the third fundamental parameter of the proposed framework. It effectively characterizes the central tendency of the data distribution and plays a pivotal role in the systematic construction and reconstruction of granules. Consequently, these three statistical metrics collectively constitute the cornerstone of the developed granulation mechanism, ensuring its representational integrity and modeling capability.

3. Building Granulation-Based Lstm for Time-Series Multi-Step Forecasting

This section elaborates on the architecture of the proposed hybrid time series forecasting model, which synergistically integrates granular computing with LSTM networks. The overall framework operates through a systematic three-phase procedure: (i) a granulation phase, where raw numerical data are mapped into statistically justifiable information granules; (ii) a prediction phase, leveraging a parallel LSTM architecture to capture the temporal dynamics of the granulated features; and (iii) a degranulation phase, dedicated to reconstructing the final predictive outputs. The comprehensive workflow of the developed framework is illustrated in Fig. 2.

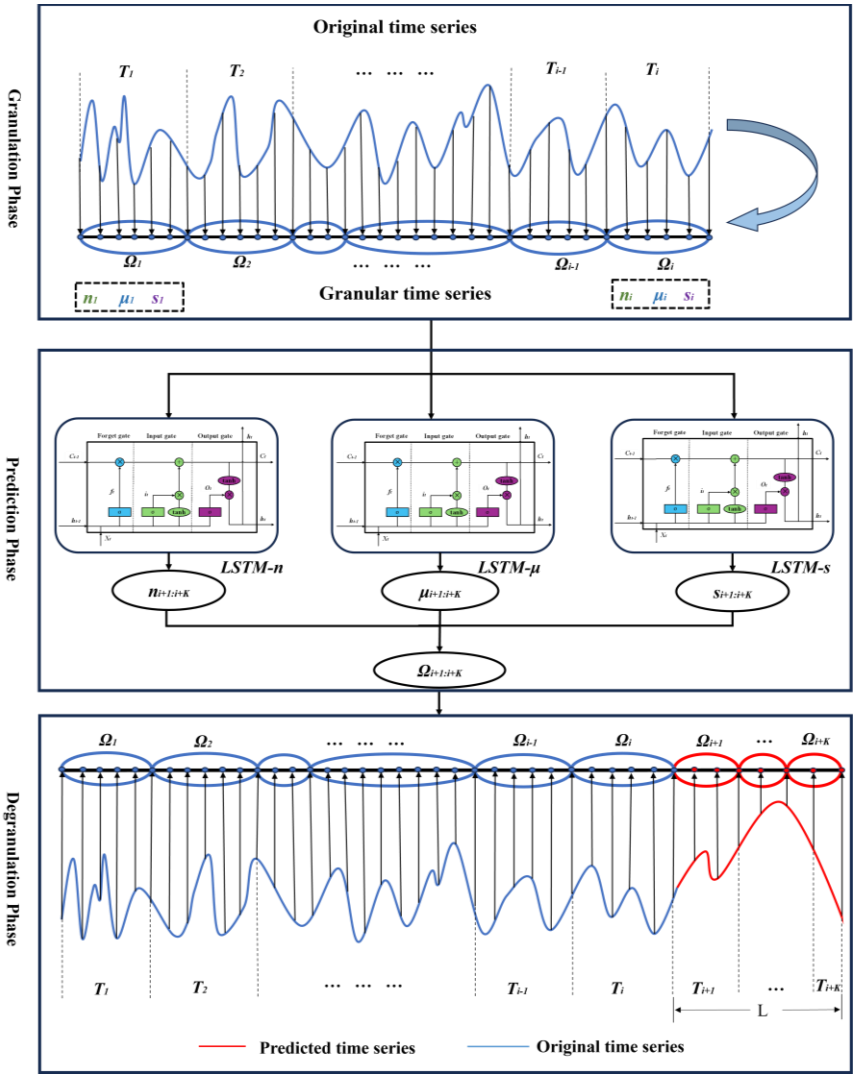

Figure 2. The complete schematic diagram of the hybrid model framework

3.1. Granulation Phase

In the granulation phase, the original numerical time series is transformed into a sequence of information granules, where each granule is represented by a statistical triplet. This transformation process strictly adheres to the principle of justifiable granularity. The granulation quality of each individual subsequence is quantitatively evaluated using the criterion defined in Equation (7). Aiming to achieve an optimal segmentation of the time series, we introduce a global objective function to maximize the cumulative granulation quality Q, as computed via Equation (8). Subsequently, a particle swarm optimization (PSO) algorithm is employed to determine the optimal partition, $\{\Omega_1, \Omega_2, \ldots, \Omega_i\}$, that maximizes the overall quality metric Q.

$$Q = Q(\Omega_1) + Q(\Omega_2) + \cdots + Q(\Omega_i) \tag{8}$$

It should be emphasized that the PSO algorithm is applied directly to the original time series to determine the optimal set of segmentation points $\{T_1, T_2, \ldots, T_i\}$, which subsequently defines the corresponding granular sequence $\{\Omega_1, \Omega_2, \ldots, \Omega_i\}$. This optimization process ensures that the resulting granules are statistically justifiable and aligned with the global objective function. The systematic execution of this granulation procedure is summarized in Algorithm 1:

Algorithm 1: Time Series Segmentation via PSO-LSTM Framework

Input:

- Raw time series $X = \{x_1, x_2, ..., x_N\}$
- Number of segments D
- Minimum spacing between segments ε_{min}
- PSO parameters: $e, m_1, m_2, N_{particles}, N_{iterations}$

Output:

- Optimal segmentation indices I
- Segmented subsequences $U = \{U_1, U_2, ..., U_i\}$

Initialization
  Initialize particle swarm P with random positions $\mathbf{p}_i \in [0,1]^N$ and zero velocities
  Evaluate initial fitness: $g(\mathbf{p}_i)$=Quality (X, Convert To Indices ($\mathbf{p}_i$, T, ε_{min}))
  Set personal best $\mathbf{p}_i^{best} \leftarrow \mathbf{p}_i$ and global best $\mathbf{g}^{best} \leftarrow \text{argmax}\, g(\mathbf{p}_i)$
PSO Optimization
for t=1 to $N_{iterations}$ **do**
  for each particle $\mathbf{p}_i \in$ P **do**
    Update velocity:
    $\mathbf{v}_i \leftarrow w\mathbf{v}_i + m_1 r_1(\mathbf{p}_i^{best} - \mathbf{p}_i) + m_2 r_2(\mathbf{g}^{best} - \mathbf{p}_i)$
    Update position: $\mathbf{p}_i \leftarrow$ Clip ($\mathbf{p}_i + \mathbf{v}_i$, [0,1])
    Convert to indices: $I_i \leftarrow$ Sort (Convert To Indices ($\mathbf{p}_i$, T, ε_{min}))
    Evaluate fitness: $g_i \leftarrow$ Quality(X,I_i)
    Update personal/global best if improved
  end for
end for

Here, e denotes the inertia weight of the particle swarm optimization algorithm, m_1 is the cognitive (individual learning) coefficient, and m_2 is the social (group learning) coefficient. ε_{min} constrains the minimum length of each granule. D represents the number of segmentation points, $N_{particles}$ is the number of particles, and $N_{iterations}$ is the maximum number of iterations. p_i denotes the normalized relative position of the particle, $g(\cdot)$ is the granulation quality evaluation function, and g^{best} represents the global best quality value. r_1 and r_2, random values drawn from the range [0, 1], are introduced to enhance the global search capability. I_i refers to the de-normalized real position of the particle.

This evaluation metric aims to determine the globally optimal segmentation of a given time series, rather than adopting a locally greedy, stepwise strategy. By emphasizing subsequences with higher in-formation density, the approach achieves a balanced trade-off between information preservation and structural compactness during granulation. As a result, it enhances both the representational capability and interpretability of the resulting granular sequence, thereby establishing a reliable foundation for sub-sequent modeling.

3.2. Prediction Phase

In the prediction phase, three dedicated LSTM networks—namely, the mean LSTM, standard deviation LSTM, and cardinality LSTM—are trained in parallel. Although these networks share the same underlying architecture, each is optimized using its corresponding input sequence. Suppose that each network generates K predicted values for the mean, standard deviation, and cardinality, respectively. The for-ecasting of these granular components is carried out according to the equations presented below:

$$n_{i+1:i+K} = f_n(LSTM_n(n_{i-w+1:i})) \tag{9}$$

$$\mu_{i+1:i+K} = f_\mu(LSTM_\mu(\mu_{i-w+1:i})) \tag{10}$$

$$\hat{s}_{i+1:i+K} = f_s(LSTM_s(s_{i-w+1:i})) \tag{11}$$

Let w denote the sliding window length and K the prediction horizon, i.e., the number of future time steps to be forecasted. The operator LSTM(·) represents the feature representation extracted through the sliding-window mechanism applied to the entire test set, whereas f(·) denotes a fully connected output layer that maps the learned representation to the prediction space. Each LSTM branch is optimized using a dedicated loss function tailored to the sta-tistical properties of its corresponding target attribute:

Cardinality LSTM Loss: Since n_t is a scalar count (often integer-valued), mean squared error (MSE) is a natural choice:

$$L_n = \frac{1}{K}\sum_{t=1}^{K}(n_t - n_t)^2 \tag{12}$$

Mean LSTM Loss: For the mean vector $\mu_t \in R^d$, a multivariate MSE is used:

$$L_\mu = \frac{1}{K}\sum_{t=1}^{K}\| \mu_t - \mu_t \|_2^2 \tag{13}$$

Standard deviation LSTM Loss: The loss function for standard deviation prediction uses MSE as well, similar to the mean branch:

$$L_s = \frac{1}{K}\sum_{t=1}^{K}(\hat{s}_t - s_t)^2 \tag{14}$$

Total Loss: The total loss is a weighted combination of the three sub-losses:

$$L_{total} = \lambda_n L_n + \lambda_\mu L_\mu + \lambda_s L_s \tag{15}$$

Where λ_n, λ_μ, λ_s are hyperparameters controlling the contribution of each branch to the overall optimization. These coefficients can be adjusted to emphasize precision in specific aspects.

Remark 2. Although the three branches share the same

fundamental LSTM architecture, their hyper-parameters are individually tailored to the statistical properties of their respective target features. For the discrete and relatively stable cardinality component, a single-layer LSTM with fewer hidden units (e.g., 25) is adopted to ensure computational efficiency. In contrast, modeling the standard deviation—which reflects fluctuation intensity and sensitivity to anomalies—demands greater representational capacity; accordingly, this branch employs more hidden units (e.g., 50). Each branch is trained independently, thereby enabling effective and non-interfering modeling of the corresponding granular components.

3.3. Degranulation Phase

The degranulation phase reconstructs the time series from the predicted granular representations. Specifically, the K predicted sets of means, standard deviations, and cardinalities obtained in the prediction phase are assembled into K forecasted granules, de-noted as Ω_{pre}:

$$\Omega_{pre} = (\mu_{pre}, s_{pre}, n_{pre}), pre = 1, ..., K \tag{16}$$

The time series values within each forecasted granule are then generated by assuming a local Gaussian distribution centered at the predicted mean μ_{pre}, with dispersion governed by the predicted standard deviation s_{pre}. For each predicted granule, a sequence of n_{pre} values is generated as follows:

$$x_{pre}^{(j)} = \mu_{pre} + s_{pre} \bullet \delta_j, \ \ j = 1, 2, ..., n_{pre} \tag{17}$$

Where j denotes the index of the predicted value within the granule, and δ_j represents a deterministic or stochastic deviation factor (e.g., sampled from N(0, 1) or derived from a predefined pattern to ensure stability and smoothness). The selection of δ_j may further incorporate historical fluctuation patterns observed in similar granules to preserve temporal consistency.

After generating the values within all forecasted granules, the reconstructed time series is obtained by concatenating the corresponding outputs, as expressed in (18):

$$X = \bigcup_{pre=1}^{K} \{ x_{pre}^{(j)} \mid j = 1, 2, ..., n_{pre} \} \tag{18}$$

This reconstruction strategy enables the model to preserve both the local statistical properties and the global temporal dynamics of the original time series. Compared with direct pointwise prediction, the granule-based reconstruction incorporating standard deviation provides greater flexibility, smoother transitions between adjacent segments, and enhanced robustness to noise and uncertainty.

3.4. Analysis of Error Sources

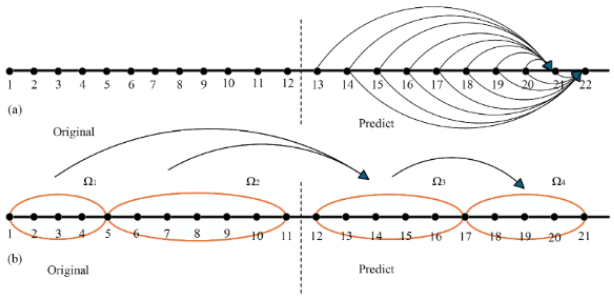

Figure 3. Error Comparison. (a) Error sources in LSTM multi-step iterative prediction. (b) Error sources in granular computing-based granule prediction

The proposed framework redefines the prediction unit by shifting from pointwise to granule-wise forecasting, thereby improving performance through information abstraction. This transformation reduces the number of required iterative steps and alleviates error propagation. As illustrated in Fig. 3(a), consider a 22-point time series. In a conventional LSTM-based multi-step forecasting strategy, predicting points 21 and 22 necessitates sequentially forecasting points 13 through 20. Such an iterative mechanism accumulates and amplifies errors, often resulting in unstable outputs. In contrast, the proposed model (Fig. 3(b)) first converts the original sequence into a granular representation $\{\Omega_1, \Omega_2\}$, where each granule summarizes the statistical characteristics of a subsequence. Based solely on the two preceding granules, the model directly predicts Ω_3 and Ω_4. Because each granule encapsulates multiple future observations, the effective iterative depth is substantially reduced, and prediction errors are confined to the granule level rather than propagated point by point. This granule-wise paradigm embodies the central principle of granular computing: reducing modeling complexity through higher-level abstraction while preserving essential structural information.

4. Experimental Studies

This section presents a comprehensive experimental evaluation of the proposed framework, covering both forecasting performance across diverse prediction tasks and the effectiveness of its internal architecture.

Experiments are conducted on multiple time series datasets to evaluate the model under varying data scales and application scenarios. Specifically, three widely used small-scale benchmark datasets—the monthly mean total sunspot series, the daily mean flow of the Oldman River, and the historical water level records of Lake Erie—are employed, together with a large-scale $PM_{2.5}$ forecasting task. This configuration facilitates a systematic evaluation on both classical benchmark datasets and more complex real-world forecasting problems.

The proposed method is compared with several representative long-horizon forecasting approaches, including AR(3), MA(3), ARIMA(3, 1, 3), a nonlinear autoregressive neural network (NAR), and LSTM, as well as the models developed by Feng [31] and Wang [29]. In addition, the results of the Probability Weight-ed Fuzzy Time Series (PWFTS) model are obtained using the publicly available Python Fuzzy Time Series library (PyFTS) [32].

Beyond predictive accuracy, further experiments are conducted to examine the robustness and internal mechanisms of the proposed forecasting framework. First, the influence of the prediction horizon is analyzed to evaluate the model's stability under different forecasting depths. Next, a granule-level error analysis is performed to investigate the residual distributions of different granular attributes and their interactions. A heatmap visualization is also provided to reveal error patterns across temporal granules and to illustrate the structural characteristics of the granular forecasting process. Finally, an ablation study compares the proposed parallel LSTM architecture with a joint LSTM model, highlighting the contribution of the parallel structure to both forecasting accuracy and computational efficiency.

In all subsequent experiments, each dataset is divided into a training set and a test set, with the first 90% of observations used for training and the remaining 10% reserved for testing. To quantitatively evaluate the proposed method, three widely

adopted performance metrics are employed: Root Mean Squared Error (RMSE), Symmetric Mean Absolute Percentage Error (SMAPE), and Mean Absolute Scaled Error (MASE), which are defined as follows:

$$RMSE = \sqrt{\frac{1}{L}\sum_{t=1}^{L}(z_{M+t} - \hat{z}_{M+t})^2} \tag{19}$$

$$SMAPE = \frac{1}{L}\sum_{t=1}^{L}\left(\frac{|z_{M+t} - \hat{z}_{M+t}|}{(|z_{M+t}| + |\hat{z}_{M+t}|)/2}\right) \tag{20}$$

To evaluate long-horizon forecasting performance, the MASE metric is adapted as follows:

$$MASE_L = \frac{1}{L}\sum_{t=1}^{L}\left(\frac{|z_{M+t} - \hat{z}_{M+t}|}{\frac{1}{L}\sum_{t=1}^{L}|z_{M+t} - z_{M+t-L}|}\right) \tag{21}$$

Where Z_{M+t} is the actual value at time $M+t$, $\hat{z}_{M+t}$ is the predicted value provided by the model, and L is the prediction horizon.

4.1. Experiments on Small-Scale Public Time Series Datasets

For one-step prediction models, multi-step forecasting typically relies on an iterative strategy: the model first predicts the value at step $n+1$, which is then fed back as input to generate the prediction for the subsequent step. This procedure continues until the desired forecasting horizon L is reached. However, iterative error accumulation is unavoidable in such frameworks and may substantially degrade long-horizon prediction performance. To more clearly demonstrate the effectiveness of the proposed approach, experiments are conducted on three bench-mark datasets: the monthly mean total sunspot series [33], the daily average flow of the Oldman River [34], and the historical water level records of Lake Erie [35]. For each dataset, the proposed model is compared with several representative traditional and modern forecasting methods.

The three LSTM branches adopt slightly different hyperparameter settings to accommodate their distinct modeling objectives. The cardinality and mean networks employ 25 hidden units and are trained for 400 epochs, while the standard deviation branch uses 50 hidden units and 600 epochs to better capture fluctuation dynamics and variability patterns. All models are trained with the Adam optimizer using a sliding window of size 5. The learning rate starts at 0.06 and is reduced by a factor of 0.2 every 125 epochs.

Tables 1–3 show that the proposed model yields the lowest RMSE across different datasets and forecasting horizons, indicating strong predictive accuracy and robustness. On the sunspot series, the proposed approach records an RMSE of 40.8235 at the 60-step horizon, yielding lower errors than LSTM (58.5824) and PWFTS (70.4001).In hydrological forecasting tasks, such as the Oldman River dataset, the model achieves an RMSE of 30.3993 at the 60-step horizon, substantially lower than AR(3) (54.4278) and LSTM (40.1845).Similarly, for the Lake Erie dataset, the proposed method consistently yields the lowest RMSE across all forecasting lengths, maintaining a value of 0.1801 even at 40 steps and outperforming classical approaches such as Holt–Winters.

Table 1. RMSE comparison for the monthly mean total sunspot time series

Method	Prediction Horizon			
	10	20	40	60
Naïve	37.8340	49.9453	77.0604	101.1822
Holt-Winters	35.9399	53.1183	89.7882	142.8540
AR (3)	32.2356	42.8007	53.0566	63.6328
MA (3)	63.7396	66.5689	67.6228	68.4253
ARIMA (3,1,3)	32.6289	44.3423	37.3471	99.3976
NARnet	34.3969	44.1515	54.3375	68.6114
LSTM	39.5332	53.8472	58.5824	72.0954
Feng's [30]	40.6037	42.1338	43.9904	52.0902
PWFTS [31]	44.8004	55.4753	70.4001	72.6932
Wang's [28]	23.2726	39.0848	39.1397	45.8642
Our approach	20.0731	33.5681	35.4936	40.8235

Table 2. RMSE comparison for the historical levels of lake erie time series

Method	Prediction Horizon			
	10	20	40	60
Naïve	0.2664	0.3026	0.3243	0.3331
Holt-Winters	0.6469	1.0248	1.6403	3.3415
AR (3)	0.2851	0.2891	0.3218	0.3191
MA (3)	0.3096	0.3249	0.3334	0.3346
ARIMA (3,1,3)	0.1950	0.2606	0.3117	0.2727
NARnet	0.2823	0.2974	0.3415	0.3006
LSTM	0.3523	0.3746	0.3674	0.3377
Feng's [30]	0.2424	0.2489	0.2744	0.2591
PWFTS [31]	0.3093	0.3093	0.3094	0.3094
Wang's [28]	0.1869	0.2448	0.1854	0.2387
Our approach	0.1774	0.2109	0.1801	0.2384

Table 3. RMSE comparison for the mean daily flow of the oldman rivertime series

Method	Prediction Horizon			
	10	20	40	60
Naïve	31.7489	36.3641	40.7552	64.2856
Holt-Winters	31.8982	36.3127	41.1088	65.9765
AR (3)	19.5275	28.5757	47.5242	54.4278
MA (3)	52.5458	56.5477	64.0841	69.0512
ARIMA (3,1,3)	27.6688	47.9785	44.1030	53.6915
NARnet	46.0986	64.4093	126.6394	86.9243
LSTM	58.2180	55.6401	56.8950	53.5308
Feng's [30]	23.2424	27.7364	33.0932	33.3773
PWFTS [31]	56.8300	73.2820	76.8872	75.8414
Wang's [28]	12.4297	26.7016	30.3304	31.4466
Our approach	10.3307	23.7859	28.3369	30.3993

4.2. Extended Experiments on Large-Scale $PM_{2.5}$ Forecasting Tasks

The evaluation is further extended to large-scale time series forecasting using $PM_{2.5}$ concentration data from four representative cities in China—Beijing, Shanghai, Shenyang, and Guangzhou [36]. These lar-ge-scale datasets enable a comprehensive ass-es-sment of the scalability and generalization capability of the proposed framework. Model performance is e-valuated under three forecasting horizons (200, 400, and 600 steps) and compared with three established approaches: ARIMA (3, 1, 3), LSTM, and the method

proposed by Wang et al. In large-scale forecasting experiments, the LSTM architecture is appropriately adjusted to accommodate increased data complexity. Specifically, the cardinality LSTM and mean LSTM are configured with 75 hidden units and trained for up to 700 epochs, whereas the standard deviation LSTM employs 100 hidden units and a maximum of 1000 training epochs to better capture volatility patterns. To effectively model long-term dependencies, a sliding-window size of 10 is adopted for all three networks. Moreover, an initial learning rate of 0.09 is uniformly applied to accelerate convergence during training.

Table 4. Performance Comparison on the $PM_{2.5}$ Dataset – Beijing

Prediction Horizon	Evaluation Metric	Method			
		ARIMA (3, 1, 3)	LSTM	Wang's	Our approach
200	RMSE	269.4253	189.2792	185.0940	171.5963
	SMAPE	1.9391	0.9129	0.8912	0.8552
	MASEL	0.9536	0.8098	0.7356	0.6991
400	RMSE	239.3077	193.8496	186.9400	165.5697
	SMAPE	1.8951	1.0387	0.9794	0.9031
	MASEL	0.7572	0.6392	0.5909	0.5257
600	RMSE	226.7751	196.5292	171.2229	151.3349
	SMAPE	1.9465	1.0357	0.9336	0.9081
	MASEL	1.1425	0.9090	0.8480	0.8895

Table 5. Performance Comparison on the $PM_{2.5}$ Dataset – Shanghai

Prediction Horizon	Evaluation Metric	Method			
		ARIMA (3, 1, 3)	LSTM	Wang's	Our approach
200	RMSE	24.2027	6.4011	4.2207	4.00392
	SMAPE	1.9562	1.4360	1.7588	1.4216
	MASEL	2.8620	0.5083	0.4148	0.4162
400	RMSE	23.3817	7.1073	5.7757	6.5497
	SMAPE	1.7970	1.3589	1.0997	0.9996
	MASEL	2.8980	0.8383	0.5993	0.8463
600	RMSE	24.7841	7.9855	7.3374	5.2344
	SMAPE	1.8643	1.5698	0.9392	0.8967
	MASEL	2.7529	0.7387	0.6477	0.6121

Table 6. Performance Comparison on the $PM_{2.5}$ Dataset – Shenyang

Prediction Horizon	Evaluation Metric	Method			
		ARIMA (3, 1, 3)	LSTM	Wang's	Our approach
200	RMSE	15.8330	8.1002	7.6563	7.2394
	SMAPE	2.0000	0.5937	0.4866	0.4227
	MASEL	2.1672	1.0467	0.9403	0.8631
400	RMSE	13.8488	8.4603	7.6507	7.4985
	SMAPE	2.0000	0.6704	0.5946	0.5665
	MASEL	1.5506	0.8771	0.7714	0.7559
600	RMSE	11.8335	8.9411	8.8314	7.5697
	SMAPE	1.9994	0.7655	0.9163	0.6635
	MASEL	1.4659	1.0806	1.0156	0.9963

Table 7. Performance Comparison on the $PM_{2.5}$ Dataset – Guangzhou

Prediction Horizon	Evaluation Metric	Method			
		ARIMA (3, 1, 3)	LSTM	Wang's	Our approach
200	RMSE	11.8622	19.2475	10.7182	10.2356
	SMAPE	1.9968	0.8338	1.5639	0.8786
	MASEL	1.4383	2.0222	1.2779	1.2031
400	RMSE	10.8488	38.3961	5.7736	11.7662
	SMAPE	1.9902	1.4717	0.5429	0.4396
	MASEL	1.4052	4.9870	0.6267	0.5637
600	RMSE	11.3305	52.7930	5.4014	13.5576
	SMAPE	1.9814	1.7652	0.4784	1.0674
	MASEL	1.2714	6.1175	0.5098	0.5001

Experimental results in Tables 4–7 for Beijing, Shanghai, Shenyang, and Guangzhou show that the proposed method outperforms baseline models in most cases. On the Beijing dataset, it achieves the lowest RMSE and MASE across all horizons, with an RMSE of 151.3349 at 600 steps. For Shanghai, it performs best at 200 steps (RMSE: 4.0039; MASE: 0.4162), while Wang's method slightly outperforms it at longer horizons. In Shenyang, the proposed model attains the lowest RMSE and SMAPE at all steps, indicating strong robustness. For Guangzhou, although Wang's model performs slightly better at 600 steps, the proposed method achieves the best results at 400 steps (SMAPE: 0.4396; MASE: 0.5637). Overall, the proposed framework demonstrates competitive and often superior performance in long-horizon forecasting across different cities. Fig. 4 further illustrates the long-term prediction waveforms for these datasets.

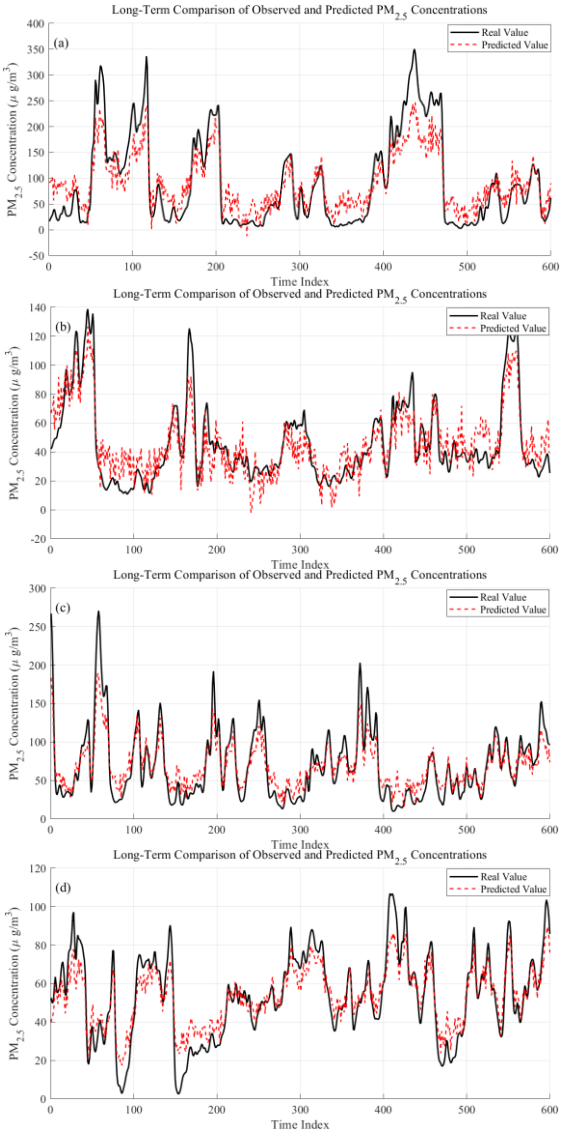

Figure 4. Long-term comparison between observed and predicted $PM_{2.5}$ concentrations in four representative cities: (a) Beijing, (b) Shanghai, (c) Shenyang, and (d) Guangzhou

As shown in Fig. 4, waveform comparisons between the observed and predicted $PM_{2.5}$ concentrations are provided for Beijing, Guangzhou, Shanghai, and Shenyang. Each subfigure corresponds to a different city, enabling a direct visual comparison of prediction performance across diverse urban environments. The predicted sequences generally follow the overall variation trends of the observed data across the 1,500-hour testing period. The model is able to reproduce most fluctuation patterns and capture several major concentration peaks. These results further demonstrate the stability and generalization capability of the proposed approach.

4.3. Sensitivity Analysis of ε_{min}

The influence of the minimum inter-granule interval ε_{min} on segmentation quality is evaluated through a dedicated sensitivity analysis. This parameter determines the minimum separation between adjacent granules and directly affects the balance between structural compactness and representation resolution. By varying ε_{min}, its impact on overall granulation quality can be systematically assessed, pro-viding practical guidance for selecting an appropriate granulation scale.

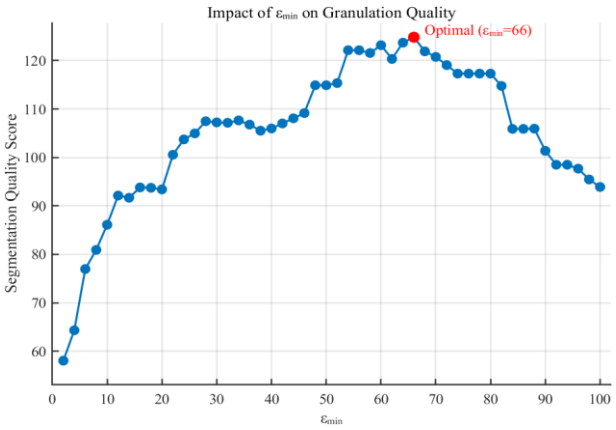

Figure 5. Impact of ε_{min} on Granulation Quality

As shown in Fig. 5, segmentation quality exhibits a clear non-monotonic trend with respect to ε_{min}. At small values, the quality score increases rapidly, indicating that overly dense granulation leads to fragmented structures and reduced representation effectiveness. When ε_{min} lies within a moderate range (approximately 50–70), the quality stabilizes at a high level, with the optimal performance observed at $\varepsilon_{min} = 66$. This suggests that an appropriate separation between granules enhances structural coherence while preserving essential temporal characteristics. However, further increases in ε_{min} lead to a gradual decline in quality, as excessively large intervals result in the loss of fine-grained information. Overall, these results reveal a clear trade-off between granulation resolution and structural compactness, highlighting the im-portance of selecting a suitable ε_{min}.

4.4. Granule-Level Error Distribution and Structural Analysis

This subsection investigates the internal behavior of the proposed model through a detailed granule-level error analysis. The prediction errors are examined from two complementary perspectives: the marginal error associated with each individual attribute and the joint error characteristics of the entire attribute set. Additionally, a granule-level error heat-map is provided to briefly illustrate the sequential distribution of prediction errors across consecutive granules, offering a supplementary view of the model's temporal stability.

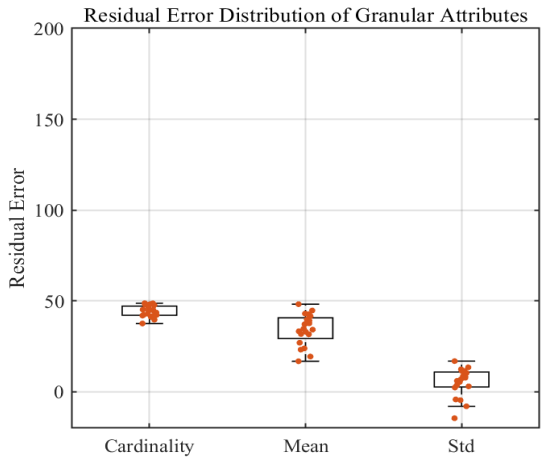

Figure 6. Residual error distributions of different granular attributes

Fig. 6 shows the residual error distributions of three granular attributes (cardinality, mean, and standard deviation) predicted by the proposed parallel framework. The residuals of cardinality are highly concentrated, with a narrow spread, indicating stable estimation of granule size. In contrast, the mean exhibits a wider dispersion, suggesting greater variability, though without evident extreme values. The residuals of standard deviation remain within a relatively small range, with both positive and negative deviations observed, indicating that prediction errors are effectively controlled despite the difficulty of modeling fluctuation intensity. Overall, the results demonstrate consistent predictive behavior across different attributes, supporting reliable structural error analysis.

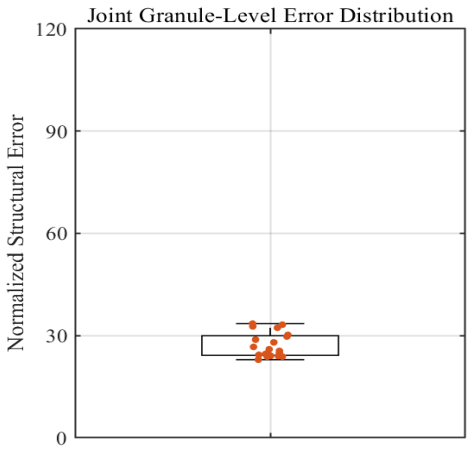

Figure 7. Joint Granule-Level Error Distribution

To further evaluate the structural stability of the proposed multi-attribute prediction framework, a joint granule-level error analysis is conducted. The normalized structural errors of cardinality, mean, and standard deviation are integrated into a unified metric to measure the overall consistency of the predicted granules. As shown in Fig. 7, most granules are concentrated within a narrow interval, with a median around 28 and a compact interquartile range. Although several scattered points indicate moderate variations among granules, the overall distribution remains bounded without evident extreme outliers. This result suggests that the simultaneous prediction of multiple granular attributes does not lead to significant structural inconsistency. Instead, the proposed framework maintains stable structural relationships among granule attributes, supporting reliable long-horizon time series forecasting.

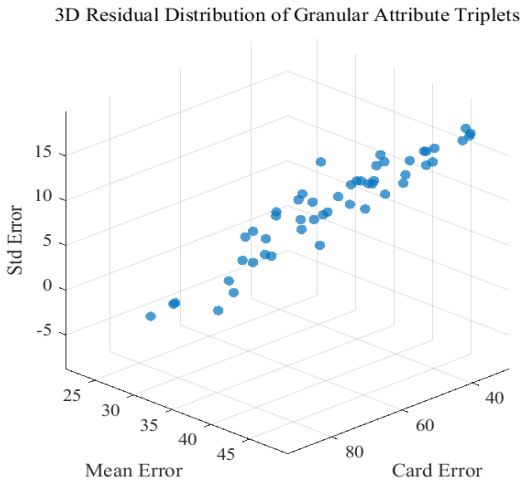

Figure 8. 3D Residual Distribution of Granular Triplet

A three-dimensional visualization is used to examine the joint residual distribution of the granular triplet (cardinality, mean, and standard deviation). Each point represents a granule and reflects the coupled prediction errors across these attributes. As shown in Fig. 8, the residuals form a compact and continuous cluster rather than scattered patterns, indicating stable model behavior. A positive correlation exists among the three error components, suggesting that increases in one error are generally accompanied by proportional changes in the others. Moreover, the absence of distant outliers implies that no anomalous granules dominate the error space. Overall, these results confirm the stability of the proposed parallel framework, with prediction errors remaining well regulated and structurally consistent.

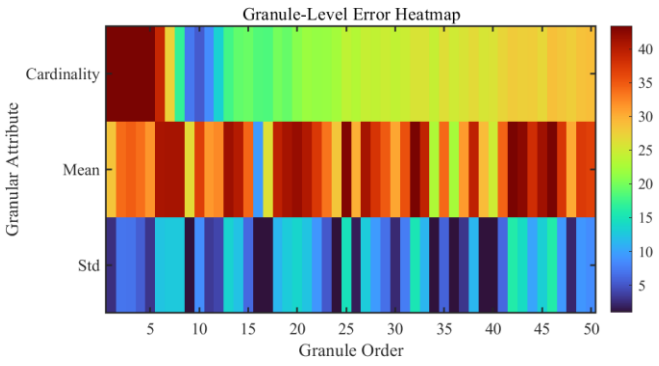

Figure 9. Granule-Level Error Heatmap

A granule-level error heatmap is used to visualize predictive performance across 50 consecutive granules for three key attributes: cardinality, mean, and standard deviation. The color gradient from deep blue (low error) to dark red (high error) provides an intuitive representation of the error distribution along the prediction horizon, as illustrated in Fig. 9. The results show that the cardinality attribute exhibits relatively higher errors in the initial granules, which gradually decrease as the granule index increases, indicating progressive stabilization of granule size estimation. In contrast, the mean attribute presents more frequent yet bounded variations, reflecting its sensitivity to underlying temporal dynamics. Mean-while, the standard deviation maintains consistently low error levels throughout the sequence, suggesting reliable modeling of fluctuation intensity. Importantly, no systematic error propagation is observed across the 50-granule horizon. This bounded error pattern indicates that the proposed parallel prediction architecture effectively mitigates long-term error accumulation, thereby supporting stable multi-attribute forecasting over

extended horizons.

4.5. Validation of the Effectiveness of the Parallel LSTM Structure

The effectiveness of the proposed parallel LSTM architecture is further examined through an ablation study comparing it with a joint LSTM configuration. In the joint setup, the three granular attributes—cardinality, mean, and standard deviation—are concatenated and fed into a single LSTM network for unified modeling. In contrast, the parallel architecture employs three independent LSTM networks to predict each attribute separately. Both configurations are evaluated under identical input data and forecasting horizons to ensure a fair comparison.

Table 8. RMSE and Time Cost Comparison between Parallel and Joint LSTM Structures

Method	Prediction Horizon					
	200		400		600	
	RMSE	Time (s)	RMSE	Time (s)	RMSE	Time (s)
Parallel LSTM	171.5963	98.3677	165.5697	102.3399	151.3349	116.0022
Joint LSTM	188.4121	115.3987	178.3488	124.9317	172.1154	150.2326

Table 8 reports the RMSE and prediction time of both architectures under different forecasting horizons (200, 400, and 600 steps). The results indicate that the parallel LSTM architecture consistently achieves higher predictive accuracy than the joint configuration, particularly in long-horizon forecasting scenarios. In addition, the parallel structure demonstrates reduced computational time across all horizons, suggesting improved efficiency without compromising accuracy. These findings highlight the practical advantages of the decoupled modeling strategy in large-scale and long-term forecasting tasks.

5. Conclusion

This study proposes a hybrid forecasting frame-work that integrates granular computing with LSTM networks to address the well-known issue of error accumulation in conventional multi-step time series forecasting. Experimental results across diverse datasets indicate that the proposed model achieves competitive and often superior performance compared with various benchmark approaches, particularly in long-horizon forecasting tasks, demonstrating strong robustness and predictive reliability.

The main contributions and strengths of the proposed framework can be summarized as follows:

(1) By transforming the original time series into granular triples consisting of cardinality, mean, and standard deviation, the framework captures essential statistical characteristics while reducing the effective modeling complexity. This granulation strategy enhances the model's capacity to represent long-term dependencies and leads to stable and accurate pre-dictions, especially under extended forecasting horizons.

(2) Replacing the covariance matrix with standard deviation significantly reduces computational complexity in univariate time series forecasting. This substitution preserves key fluctuation dynamics while avoiding redundant computation during feature extraction. The resulting efficiency gains become particularly evident in large-scale datasets, where the framework maintains strong predictive performance with reduced computational overhead.

(3) The parallel LSTM architecture improves both training efficiency and predictive stability. By assigning each statistical component—cardinality, mean, and standard deviation—to an independent LSTM branch, the framework decouples feature learning, mitigates cross-attribute interference, and enhances structural interpretability. Ablation experiments further indicate that the parallel architecture yiel-ds lower RMSE values and more stable performance across various forecasting horizons when compared with a unified LSTM configuration.

Despite these advantages, certain limitations remain. The current granulation mechanism does not fully model intervariable dependencies in multivariate time series. Future work will investigate more expr-essive granulation strategies to better model cross-variable interactions. The framework will also be extended to high-dimensional forecasting tasks to improve predictive accuracy and adaptability.

Acknowledgments

Henan Provincial Key Scientific Research Projects Plan for Higher Education Institutions [25A120004]

Natural Science Foundation of Henan Province [252300423319]

Data Availability Statement

The datasets used in this study are publicly available from the following sources:

The sunspot dataset is publicly accessible through the Solar Influences Data Analysis Center (SIDC), which maintains the official data repository.

Lake Erie water level data were obtained from the US Army Corps of Engineers via the Great Lakes water level dashboard, source as cited by GLISA (Great Lakes Integrated Sciences and Assessments).

Oldman River dataset: Rob J Hyndman's Time Series Data Library, available a http://robjhynd-man.com/tsdldata/astatkie/oldman.dat.

PM2.5 data of five Chinese cities: The dataset is publicly available from the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/PM2.5+Data+of+Five+Chinese+Cities.

All datasets are open-access and were used solely for academic research in this work.

CRediT Authorship Contribution Sta-Tement

Zhitao Jia: Conceptualization, Investigation, Methodology, Software, Writing – original draft.

Lijie Jia: Funding acquisition, Writing- review & editing.

Lianchao Qiu: Funding acquisition, Supervision, Writing – review & editing.

References

[1] G. Liu, F. Xiao, C. T. Lin, and Z. Cao, “A fuzzy interval time-series energy and financial foreca-sting model using network-based multiple time-frequency spaces and the induced-ordered weighted averaging aggregation ope-ration,” IEEE Trans. Fuzzy Syst., vol. 28, no. 11, pp. 2677–2690, Nov. 2020.

[2] N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, “Deep adaptive input normalization for time series forecasting,” IE-EE Trans. Neural. Netw. Learn. Syst., vol. 31, no. 9, pp. 3760–3765, Sep. 2020.

[3] P. C. de Lima, H. J. Silva, R. S. Ballini, and F. G. Guimarães, “Probabilistic forecasting with fuzzy time series,” IEEE Trans. Fuzzy Syst., vol. 28, no. 8, pp. 1771–1784, Aug. 2020.

[4] E. Isufi, A. Loukas, N. Perraudin, and G. Leus, “Forecasting time series with VARMA recu-rsions on graphs,” IEEE Trans. Signal Process., vol. 67, no. 18, pp. 4870–4885, Sep. 2019.

[5] J. Wen, J. Yang, B. Jiang, H. Song, and H. Wang, “Big data driven marine environment infor-mation forecasting: A time series prediction network,” IEEE Trans. Fuzzy Syst., vol. 29, no. 1, pp. 4–18, Jan. 2021.

[6] R. R. Sharma, M. Kumar, S. Maheshwari, and K. P. Ray, “EVDHM-ARIMA-based time series forecasting model and its application for COVID-19 cases,” IEEE Trans. Instrum. Meas., vol. 70, 2021, Art no. 6502210.

[7] H. Khani and H. E. Z. Farag, “An online-calibrated time series based model for day-ahead natural gas demand forecasting,” IEEE Trans. Ind. Inform., vol. 15, no. 4, pp. 2112–2123, Apr. 2019.

[8] S. Jahandari, A. Kalhor, and B. N. Araabi, “Online forecasting of synchronous time series based on evolving linear models,” IEEE Trans. Syst. Man Cybern., Syst., vol. 50, no. 5, pp. 1865–1876, May 2020.

[9] J. P. González, A. M. S. M. San Roque, and E. A. Pérez, “Forecasting functional time series with a new Hilbertian ARMA Xmodel: App-lication to electricity price forecasting,” IEEE Trans. Power Syst., vol. 33, no. 1, pp. 545–556, Jan. 2018.

[10] G. Zhang et al., “Forecasting time series albedo using NARnet based on EEMD decomposition,” IEEE Trans. Geosci. Remote Sens., vol. 58, no. 5, pp. 3544–3557, May 2020.

[11] G. Box, G. Jenkins, Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco, 1976.

[12] M.J. Duran, D. Cros, J. Riquelme, Short-term wind power forecast based on ARX models, J. Energy Eng. 133 (3) (2007) 172–180.

[13] S.S. Pappas, L. Ekonomou, D.C. Karamo-usantas, G. Chatzarakis, S. Katsikas, P. Liatsis, Electricity demand loads modeling using AutoRegressive Moving Average (ARMA) models, Energy 33 (9) (2008) 1353–1360.

[14] E. Erdem, J. Shi, ARMA based approaches for forecasting the tuple of wind speed and direction, Appl. Energy 88 (4) (2011) 1405–1414.

[15] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, Time Series Analysis: Forecasting and Control, 4th edition, John Wiley & Sons, New York, 2008.

[16] A.I. McLeod, W.K. Li, Diagnostic checking ARMA time series models using squared-residual to correlations, J. Time Ser. Anal. 4(4) (1983) 269–273.

[17] T.A. Jilani, S.M.A. Burney, M-factor high order fuzzy time series forecasting for road accident data: analysis and design of intelligent systems using soft computing techniques, Adv. Soft Comput. 41 (2007) 246–254.

[18] C.J. Lu, T.S. Lee, C.C. Chiu, Financial time series forecasting using independent co-mponent analysis and support vector regression, Decis. Support Syst.47 (2) (2009) 115–125.

[19] Ö. Baydaroglu, K. Koçak, SVR-based pred-iction of evaporation combined with chaotic approach, J. Hydrol. 508 (2014) 356–363.

[20] Shakeel Ahmad, Jinsong Tao, and Rahat Ali, "Optimizing Short-term Power Load Fore-casting with a Hybrid PSO-SVR-LSTM App-roach based on Data Analysis," Engineering Letters, vol. 33, no. 7, pp2501-2513, 2025.

[21] Ze Gong, and Yong Zhang, "LSTM Short-term Electricity Load Forecasting Model Based on Improved Sparrow Search Algorithm and Ensemble Learning," Engineering Letters, vol. 33, no. 8, pp2996-3014, 2025.

[22] Yang Xu, Junyan Hu, Cheng Yao, Xiaorui Hu, Hexin Peng, and Hongyu Long, "Integrating CEEMDAN-EWT and Hybrid Neural Network for Traffic Flow Forecasting," Engineering Letters, vol. 33, no. 8, pp3025-3039, 2025.

[23] Jia He, Changfeng Zhu, Runtian He, Yunqi Fu, and Jie Wang, "Research on Short-time Inbound Passenger Flow Prediction of Urban Railways Considering Station Flow Het-erogeneity," Engineering Letters, vol. 33, no. 9, pp3425-3439, 2025.

[24] A. Tugˇrul Seyhan, Gökmen Tayfur, Murat Karakurt, Metin Tanogl’u, Artificial neural network (ANN) prediction of compressive strength of VARTM processed polymer com-posites Original, Comput. Mater. Sci. 34 (1) (2005) 99–105.

[25] R. Al-Hmouz, W. Pedrycz, and A. Balamash, “Description and prediction of time series: a general framework of granular computing,” Expert Syst. Appl., vol. 42, no. 10, pp. 4830–4839, Jun. 2015.

[26] W. Froelich and W. Pedrycz, “Fuzzy cognitive maps in the modeling of granular time series,” Knowl.-Based Syst., vol. 115, pp. 110–122, Jan. 2017.

[27] X. Yang, F. Yu, and W. Pedrycz, “Long-term forecasting of time series based on linear fuzzy information granules and fuzzy inference system,” Int. J. Approx. Reasoning, vol. 81, pp. 1–27, Feb. 2017.

[28] Y. Tang, F. Yu, W. Pedrycz, X. Yang, J. Wang and S. Liu, "Building Trend Fuzzy Granulation-Based LSTM Recurrent Neural Network for Long-Term Time-Series Forecasting," in IEEE Transactions on Fuzzy Systems, vol. 30, no. 6, pp. 1599-1613, June 2022, doi: 10.1109/TFU-ZZ.2021.3062723.

[29] W. Wang, W. Liu and H. Chen, "Time-Series Forecasting via Fuzzy-Probabilistic Approach With Evolving Clustering-Based Granulation," in IEEE Transactions on Fuzzy Systems, vol. 30, no. 12, pp. 5324-5336, Dec. 2022, doi: 10.110-9/TFUZZ.2022.3173684.

[30] W. Pedrycz and G. Vukovich, “Abstraction and specialization of information granules,” IEEE Trans. Syst. Man Cybern. B, Cybern., vol. 31, no. 1, pp. 106–111, Feb. 2001.

[31] G. L. Feng, L. Y. Zhang, J. H. Yang, and W. Lu, “Long-term prediction of time series using fuzzy cognitive maps,” Eng. Appl. Artif. Intel., vol. 102, 2021, Art. no. 104274.

[32] P. C. de Lima, H. J. Silva, R. S. Ballini, and F. G. Guimarães, “Probabilistic forecasting with fuzzy time series,” IEEE Trans. Fuzzy Syst., vol. 28, no. 8, pp. 1771–1784, Aug. 2020.

[33] Sunspot index and long-term solar observations. Accessed: May 17, 2022.[Online]. Available: http://www.sidc.be/silso/datafiles.

[34] Historical Lake Levels- Lake Erie. Accessed: May17, 2022. [Online]. Available: http://www.glisaclimate.org/projects/484/page/2536.

[35] Mean Daily Flow Oldman River. Accessed: May 17, 2022. [Online]. Available: http://robjh-yndman.com/tsdldata/astatkie/oldman.dat.

[36] UCI machine learning repository: Data sets. Accessed: May17, 2022. [Online]. Available: https://archive.ics.uci.edu/dataset/394/pm2+5+data+of+five+chinese+cities.