World Journal of Information Technology
Print ISSN: 2959-9903
Online ISSN: 2959-9911
DOI: https://doi.org/10.61784/wjit3058

MULTI-GRANULARITY TIME SERIES FORECASTING METHODS BASED ON DUAL-CHANNEL FUSION

XueYuan Zhu*, JiaXin Peng
School of Transportation, Changsha University of Science & Technology, Changsha 410114, Hunan, China.
Corresponding Author: XueYuan Zhu, Email: xueyuan_zhu6007@163.com

Abstract: In high-frequency data environments, traditional time-series forecasting methods generally face two major challenges. First, the structures of these models are too simple to capture both the long-term trends and short-term disturbances. Second, the forecasting granularity is too coarse to meet the refined requirements for real-time dynamic decision-making. To address these issues, this study proposes a dual-channel fusion forecasting framework, the Dual-Resolution Adaptive Forecasting Topology (DRAFT) architecture. The architecture comprises two modules: a trend modeling module and a disturbance modeling module. The modules are responsible for processing the linear trend components and nonlinear fluctuation signals in the time series data. They achieved adaptive integration of the prediction results using a lightweight fusion mechanism. Experiments on real-world datasets demonstrated that the DRAFT architecture significantly outperformed traditional single-model approaches in terms of metrics such as mean squared error (MSE) and mean absolute error (MAE), with error reductions exceeding 54.05% in certain scenarios. Furthermore, DRAFT possesses the capacity to refine the prediction output granularity to the 10-minute level, thereby providing more actionable prediction information for high-timeliness scenarios. This study establishes a new paradigm for the precise prediction of complex time-series data and provides theoretical and practical references for the construction of modular prediction systems.
Keywords: Multi-granularity prediction; Time series modeling; Model fusion; Predictive granularity refinement

1 INTRODUCTION

Among numerous real-time decision support systems, accurate predictions of future quantities are a core prerequisite for ensuring system efficiency and rational resource allocation. With the continuous advancement of data collection technology, the temporal granularity of data acquisition has become increasingly refined. However, the temporal resolution of predictive models still lags behind the requirements of real-world applications. Such issues are particularly pronounced in scenarios characterized by task-intensive scheduling and the need to respond to instantaneous fluctuations, where the requirements for the response speed and accuracy of the predictive methods are significantly heightened.

Multiscale time series simultaneously exhibit linear trends, nonlinear disturbances, and random fluctuations, which pose challenges for single modeling strategies. Traditional statistical models (e.g., ARIMA) excel at handling stable trends but struggle to capture high-frequency nonlinear changes, with a limited ability to fit complex nonlinear features [1-2]. In contrast, while deep learning models (such as LSTM) possess strong nonlinear modeling capabilities, they often exhibit limitations in terms of interpretability, stability, and the handling of short-term anomalies [3-4]. Additionally, most current forecasting research remains focused on hourly or daily granularity, with a coarse temporal resolution that fails to meet the practical demand for “minute-level” dynamic responses, creating a significant tension between timeliness and practicality.

Given the dual challenges of structural adaptability and time sensitivity in multi-granularity time-series forecasting tasks, there is an urgent need for a hybrid forecasting framework that can balance trend modeling, sudden change capture, and multi-timescale response [5]. This study proposes a modular, responsive, and scalable structured forecasting system called Dual-Resolution Adaptive Forecasting Topology (DRAFT). This architecture builds a multifunctional collaborative mechanism, enabling the model to simultaneously capture long-term trends and short-term fluctuations, effectively bridging the performance gap between static modeling and dynamic response. Its core concept is to use ARIMA to capture the linear patterns in the data while using LSTM to learn the complex nonlinear structures in the residuals. Finally, through a fusion mechanism, the outputs of both models are balanced to construct a prediction system with both robustness and generalization capabilities.

Unlike existing single-strategy approaches, the DRAFT architecture significantly enhances the generalization capabilities of the system while ensuring structural transparency through hierarchical learning and output fusion mechanisms. Its design emphasizes a fine-grained response in prediction granularity, offering comprehensive adaptability from macro-level trend analysis to micro-level disturbance resolution, and is particularly suited for real-time prediction scenarios involving high-density time series. The DRAFT architecture innovatively integrates the advantages of multiple models, advancing the prediction granularity from the traditional hourly level to a 10-minute level, significantly enhancing the spatiotemporal adaptability of the model. Extensive experimental validation of classic multivariate time series tasks demonstrates that the system outperforms the baseline methods on multiple key evaluation metrics while maintaining good interpretability and scalability, achieving a deep integration of theory and practice. It can serve as a general-purpose solution for high-frequency prediction.

2 LITERATURE REVIEW

In the field of time-series forecasting, researchers have long explored model structures, data characteristics, and granularity response capabilities. This paper reviews existing research from the following three perspectives: (1) trend-driven modeling strategies, (2) nonlinear structure learning methods, and (3) the evolution of multi-granularity forecasting frameworks.

2.1 Trend Modeling and Robustness of Statistical Methods

Traditional time series analysis methods are primarily based on statistical modeling, with their core advantages being strong parameter interpretability and transparent modeling processes, which are particularly suitable for handling stationary sequences and linear trends. Such methods typically rely on differencing, autoregression, and error structures to construct predictive functions, with representative works including differenced moving average models under the assumption of stationarity and seasonal trend analysis tools. Bichescu et al. proposed an innovative time-series analysis method that simplifies the construction process of ARIMA models and may improve the efficiency and accuracy of predictions [6]. Li et al. used an autoregressive integrated moving average (ARIMA) model to predict the development trend of gonorrhea, providing a reference for formulating corresponding prevention and control strategies [7].

However, traditional statistical modeling methods have obvious limitations. When faced with the nonlinear disturbances and structural changes commonly found in the real world, their model architecture, based on linear assumptions and stationarity premises, struggles to effectively capture the complex dynamic changes in data, leading to a significant decline in prediction accuracy. In high-frequency, non-stationary data scenarios, such as minute-level stock price fluctuations in financial markets or real-time changes in power load, traditional models fail to promptly capture the instantaneous fluctuations and structural changes in data, resulting in severe degradation of model performance. Additionally, traditional statistical models typically use hourly or daily time windows, which have a relatively coarse temporal resolution. This fails to meet the urgent needs of modern precision decision-making systems for high-timeliness and high-accuracy predictions and cannot provide timely and accurate information support for dynamic decision-making.

2.2 Nonlinear Pattern Learning and Neural Prediction Mechanisms

In recent years, the rapid development of deep learning technology has led to revolutionary breakthroughs in the field of time-series forecasting. Neural network models oriented toward sequence modeling have demonstrated strong fitting capabilities and generalization potential in time-series forecasting. In particular, recurrent structures such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) have effectively addressed the gradient vanishing and long-term dependency issues inherent in traditional Recurrent Neural Networks (RNNs) by introducing gating mechanisms and are widely applied in nonlinear sequence modeling tasks. Ma et al. utilized LSTM neural networks to process highly nonlinear, dynamic, and time-dependent sequence data in industrial processes, providing an effective soft sensor technology for modeling issues related to the strong time-varying characteristics of the process and predicting key variables [8]. Yin et al. proposed an LSTM-based multistate vector sequence-to-sequence model for rainfall-runoff modeling, achieving a multistep runoff prediction [9].

These methods excel in capturing long-term dependencies, local anomalies, and non-stationary features, making them suitable for constructing nonlinear mappings between complex inputs and outputs. However, neural network-based methods face significant challenges. Their “black box” nature makes it difficult to intuitively interpret the internal decision-making mechanisms of the model, resulting in poor model interpretability; the training process heavily relies on large-scale labeled data, resulting in significantly reduced model performance in data-scarce scenarios; additionally, when faced with sudden short-term temporal changes, owing to the inherent delay in the model's computation and update mechanisms, neural networks often struggle to provide timely and accurate predictive responses, limiting their application in high-timeliness decision-making scenarios.

2.3 Development of Multi-Granularity Response Mechanisms and Fusion Frameworks

To balance model stability and expressive power, the academic community has increasingly turned to research on model fusion and structural integration in recent years. These methods typically integrate sub-models with different modeling properties into a unified framework to achieve hierarchical learning and prediction of different signal components. Typical fusion strategies include weighted combination, residual stacking, and hierarchical recursion, which can improve the model adaptability and prediction accuracy to a certain extent. Meanwhile, some studies have attempted to introduce prediction tasks with finer temporal granularity (e.g., 10-minute intervals) to address the dual demands for timeliness and accuracy in real-time optimization and dynamic scheduling scenarios. Lu et al. proposed an integrated multi-temporal granularity deep learning prediction method (Mul-DesLSTM) for short-term passenger flow prediction in urban rail transit systems. This method aims to address the issue of high-resolution data generated by automatic fare collection (AFC) systems being wasted [10]. He et al. proposed a dynamic multi-fusion spatiotemporal graph neural network for multivariate time-series prediction. This method aims to simultaneously capture hidden temporal and spatial patterns in spatiotemporal data [11].

To balance model stability and expressive power, the academic community has increasingly turned to research on model fusion and structural integration in recent years. These methods typically integrate sub-models with different modeling properties into a unified framework to achieve hierarchical learning and prediction of different signal components. Typical fusion strategies include weighted combination, residual stacking, and hierarchical recursion, which can improve the model adaptability and prediction accuracy to a certain extent. Meanwhile, some studies have attempted to introduce prediction tasks with finer temporal granularity (e.g., 10-minute intervals) to address the dual demands for timeliness and accuracy in real-time optimization and dynamic scheduling scenarios. Lu et al. proposed an integrated multi-temporal granularity deep learning prediction method (Mul-DesLSTM) for short-term passenger flow prediction in urban rail transit systems. This method aims to address the issue of high-resolution data generated by automatic fare collection (AFC) systems being wasted [10]. He et al. proposed a dynamic multi-fusion spatiotemporal graph neural network for multivariate time-series prediction. This method aims to simultaneously capture hidden temporal and spatial patterns in spatiotemporal data [11].

However, current related research still faces limitations, such as strong structural closedness, weak granularity adaptation mechanisms, and coarse fusion strategies. Strong structural closedness manifests as a lack of systematic construction of multi-module collaborative mechanisms, leading to insufficient information interaction and collaboration efficiency between sub-models. Weak granularity adaptation mechanisms are manifested in the widespread use of fixed-interval segmentation modeling, which struggles to dynamically respond to actual temporal changes. Coarse fusion strategies refer to the fact that most studies still rely on simple averaging or linear weighting, neglecting the heterogeneous relationships between model outputs. These limitations highlight the urgent need for a more flexible, adaptive, and interpretable fusion framework to systematically coordinate multi-module interactions, dynamically adapt to changes in granularity, and optimize fusion strategies based on output features.

2.4 Positioning and Innovation of This Study

Based on the aforementioned research, this study proposes a dual-channel adaptive forecasting architecture for multi-granularity time series, namely, Dual-Resolution Adaptive Forecasting Topology (DRAFT). This architecture decouples the linear trend components and nonlinear volatility features of time series, constructing a collaborative processing framework comprising a trend modeling module (based on an ARIMA-based linear feature extractor) and a disturbance modeling module (based on an LSTM-based nonlinear residual learner). It achieves the dynamic calibration of dual-path outputs through a lightweight voting fusion mechanism, enabling hierarchical modeling capabilities for complex data structures.

Compared with existing methods, the innovations of the DRAFT architecture are reflected in the following three aspects: (1) Clear division of functional modules, enabling each substructure to focus on specific signal component modeling tasks. The decoupled design of linear trend modeling and nonlinear disturbance learning avoids the feature interaction interference issues in traditional hybrid models, enhancing modeling efficiency while improving model interpretability; (2) Fine-grained temporal response: Breaking through the temporal granularity limitations of traditional prediction models, the DRAFT architecture introduces a temporal granularity calibration mechanism to refine prediction output granularity to the 10-minute level, significantly enhancing the model's dynamic response capability to high-frequency data; (3) Reconstructable prediction results: Supports flexible reconstruction and combination of trend and disturbance components based on actual application scenarios, forming customized prediction result output modes, effectively enhancing the model's practicality and scalability in multi-scenario decision support systems.

3 METHODOLOGY

The proposed Dual-Resolution Adaptive Forecasting Topology (DRAFT) architecture aims to effectively improve the accuracy, stability, and response speed of time-series forecasting through multi-module collaboration and structural fusion mechanisms. The architecture consists of three core components: a trend modeling module (Trend Module), disturbance capture module (Disturbance Module), and result Fusion Module (Fusion Module). The overall process is illustrated in Figure 1.

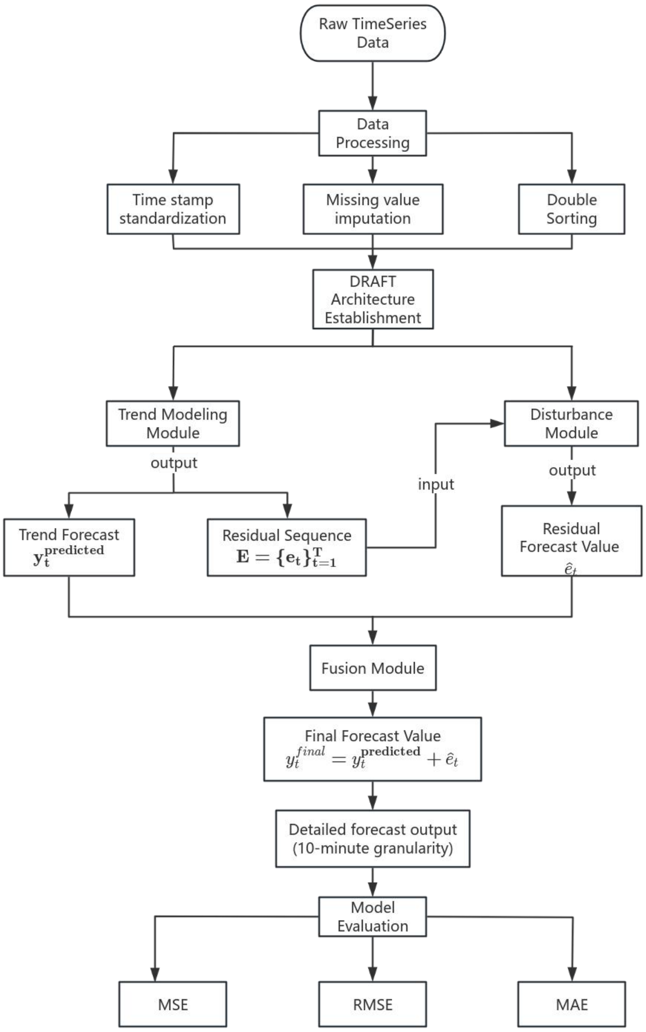

Figure 1 Overall Flow Chart

3.1 Data Preprocessing and Modeling Fundamentals

In high-frequency time-series forecasting scenarios, data preprocessing is a critical foundational step in modeling. For multidimensional datasets comprising timestamps, category identifiers, and target variables, the following standardized processing steps must be executed: initially, time-related features undergo normalization and integration, whereby discrete date and time information is consolidated into a unified timestamp format and subsequently arranged in chronological order. This ensures the consistency of the temporal logic in the data. Second, statistical interpolation methods were employed to address issues related to missing data, with the objective of ensuring data integrity and preventing modeling biases. The dataset was then sorted based on both category identifiers and time dimensions to construct a structured temporal feature matrix. This process clearly reveals the trend, periodicity, and abnormal fluctuation patterns of the target variable over time. The result is a standardized input for subsequent multi-granularity modeling.

The preprocessing framework is applicable to various types of multidimensional data with temporal dependencies. The integration of data formats, rectification of data defects, and augmentation of temporal characteristics serve as the basis for the effective training and precise prediction of multi-granularity time-series forecasting models.

3.2 Trend Modeling Module

The present module is predicated on the difference integrated moving average autoregressive model (ARIMA) and aims to analyze the linear trend components and cyclical patterns in time series, thereby modeling relatively stable long-term trends and repetitive patterns. The core process of the system under investigation revolved around the three core steps of the ARIMA model. The configuration of the ARIMA model is shown in Figure 2.

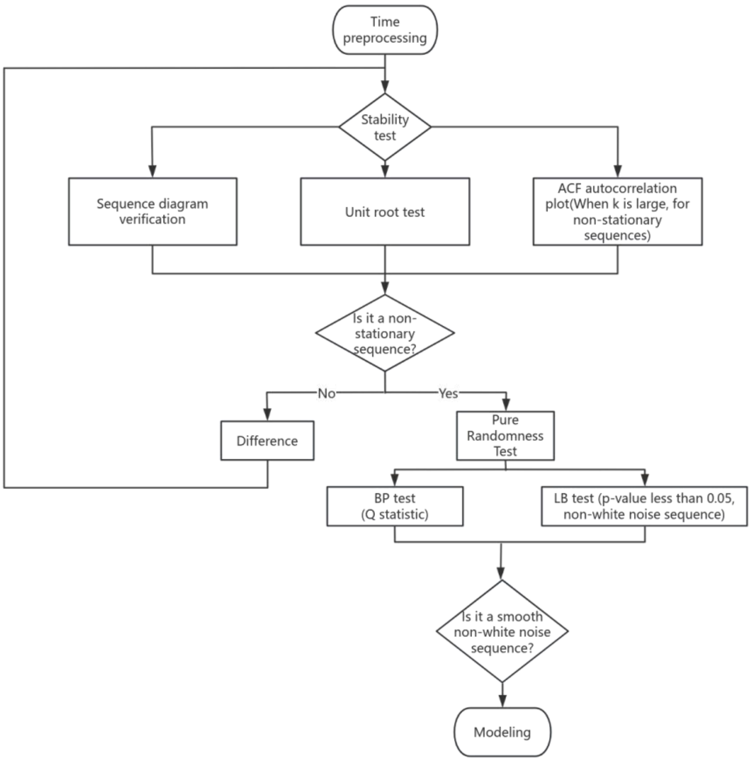

Figure 2 ARIMA Model Structure Diagram

The specific implementation logic is as follows:

3.2.1 Sequence stability analysis and differential processing

The stationarity of a time series is a prerequisite for applying the ARIMA model. The module initially employs a unit root test to ascertain the stationarity of the original series data. In instances where the series under consideration exhibits a substantial trend or seasonality, the trend component is extirpated through the implementation of differencing, thereby yielding a stationary series.

The formula for first-order differencing is as follows.

$$\Delta y_t = y_t - y_{t-1} \tag{1}$$

The formula for second-order difference is.

$$\Delta^2 y_t = \Delta y_t - \Delta y_{t-1} \tag{2}$$

In the context of a time series characterized by a linear trend, the trend can be eliminated through the implementation of first-order differencing, thereby rendering the series stationary.

The foundation for subsequent model fitting is laid by repeated differencing until the series satisfies a stationarity condition.

3.2.2 ARIMA model structure construction

Subsequent to the smoothing process, the module constructs a model based on autoregressive (AR) and moving average (MA) structures.

The autoregressive (AR) model posits that the value at the present moment y_t can be represented by a linear combination of the values at past moments and an error term.

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t \tag{3}$$

Among them, c is the constant term, ϕ_i is the autoregressive coefficient, and ϵ_t is the white noise error term.

The moving average (MA) model assumes that the current value y_t can be represented by a linear combination of the current error term and the error terms from the previous p moments.

$$y_t = \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} \tag{4}$$

The model captures the historical dependencies of the sequence through the AR term and fits the moving average pattern of the error term through the MA term, thereby achieving joint modeling of linear trends and cyclical components.

3.2.3 Hyperparameter optimization and model training

The module uses a grid search algorithm to optimize the hyperparameters of the ARIMA model (p is the autoregressive order, d is the difference order, and q is the moving-average order). The mean square error (MSE) was used as the objective function, and the optimal parameter combination was determined by minimizing the prediction error on the training set. The objective function expression is as follows:

$$\mathrm{MSE}=\frac{1}{n}\sum_{i=1}^{n}\left(y_t^{\mathrm{actual}}-y_t^{\mathrm{predicted}}\right)^2 \tag{5}$$

Where is the predicted value of the y trend module and n is the number of training samples. By iterating through the feasible combinations of (p, d, q), the parameter combination that minimizes the MSE was selected as the final model configuration.

After parameter optimization, the ARIMA model can generate a trend prediction sequence that describes the path of macro changes. This sequence removes the nonlinear disturbance components in the original data and focuses on describing the long-term trends and cyclical patterns of the data, providing residual input for the subsequent disturbance capture module.

3.3 Disturbance Module

This module is dedicated to mining nonlinear dynamic features in time series trend residuals, focusing on modeling short-term sudden changes, historical sequence dependencies, and complex fluctuation patterns to improve the model accuracy in capturing unpredictable disturbances and its dynamic response capabilities. The functionality of this module was achieved using a long short-term memory (LSTM) network model.

The core unit of the LSTM model is illustrated in Figure 3.

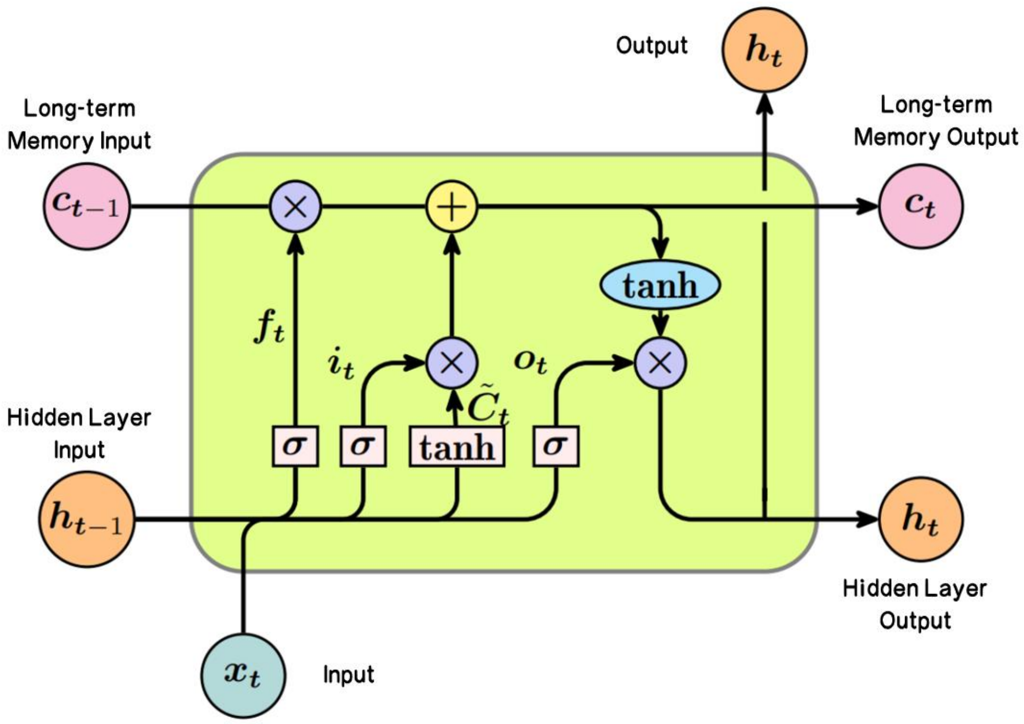

Figure 3 LSTM Model Structure Diagram

The calculation process consists of the following steps:

ForgetGate:

The ForgetGate determines how much information in memory unit C_{t-1} needs to be forgotten at the previous moment. The calculation formula is as follows:

$$f_t=\sigma(W_f\cdot[h_{t-1},x_t]+b_f) \tag{6}$$

Where σ is the sigmoid function, whose output range is 0 to 1. W_f is the weight matrix of the ForgetGate, $[h_{t-1},x_t]$ represents concatenating the hidden state h_{t-1} from the previous time step with the current input x_t, and b_f is the bias of the ForgetGate. When f_t approaches 0, it indicates that most of the information is forgotten; when f_t approaches 1, it indicates that most of the information is retained.

InputGate:

The InputGate determines the amount of new information added to the memory unit at the current moment.

First, the output of the InputGate is calculated using formula (7), while candidate memory units are produced.

$$i_t=\sigma(W_i\cdot[h_{t-1},x_t]+b_i) \tag{7}$$

The output range of function tanh is -1 to 1.

$$\tilde{C}_t=\tanh(W_C\cdot[h_{t-1},x_t]+b_C) \tag{8}$$

Then, the output of the InputGate is multiplied by the candidate memory unit to obtain the new information to be added to the memory unit.

Memory unit update: The formula for updating memory units is as follows：

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{9}$$

Where represents element-wise multiplication. This formula represents adding the information to be retained in the memory unit at the previous moment and the new information at the current moment to obtain the memory unit at that time.

OutputGate:

The OutputGate determines which information in the memory unit will be used to generate the output at the current moment. The calculation formula is as follows：

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{10}$$

The hidden state at the current time is $h_t = o_t * \tanh(C_t)$. Through the control of the output gate, the LSTM unit can selectively output information from the memory unit.

The specific implementation logic of the Disturbance Module is as follows:

3.3.1 Input design

The module input is the residual sequence $E = \{e_t\}_{t=1}^{T}$ generated by the trend module,$e_t = y_t - y_t^{predicted}$,$y_t$ is the original time series value, and $y_t^{predicted}$ is the predicted value of the trend module.Residual sequences typically contain nonlinear components that are not explained by linear trends in the original data and need to be further modeled using nonlinear models.

3.3.2 LSTM network architecture

The module adopts a multi-layer LSTM network structure, which uses its gating mechanism (InputGate, ForgetGate, OutputGate) to selectively retain historical information to capture long-range dependencies. The network structure specifically includes an embedding layer, LSTM layers, and fully connected layers. The embedding layer standardizes the input residual sequence to improve training stability; the LSTM layer stacks 2–3 layers of LSTM units, each containing n_h memory units, using the forget gate to filter out irrelevant historical information and transmitting long-term dependency features through cell states; the fully connected layer maps the hidden states output by the LSTM layer to the predicted values $\hat{e}_t$ via linear transformation, i.e., the predicted values of the perturbation components.

3.3.3 Hyperparameter optimization mechanism

Automated tuning of network hyperparameters using a random search algorithm. The core optimization parameters include: Historical window length L, which determines the time span of the input sequence, i.e., the past L residual values input into the model each time, used to capture local dependency patterns; The number of LSTM units is n_h, which controls the network's nonlinear fitting capability. A larger number of units can capture more complex feature interactions, but overfitting must be avoided; Training epochs are determined through cross-validation to prevent underfitting or overfitting; The learning rate uses an adaptive learning rate algorithm to dynamically adjust the update step size, accelerating convergence.

3.3.4 Time sliding window mechanism

To enhance the model's sensitivity to local changes, input data is processed using a nested sliding window structure. Overlapping windows cover the entire time period, enabling the model to capture local features at different time offsets and improving its responsiveness to short-term sudden changes. Mathematically, each window corresponds to a local time series segment, and its output is the residual prediction value $\hat{e}_{i+L}$ for future time points Δt, forming a “many-to-one” prediction model.

This module effectively compensates for the blind spots of the trend module in processing non-stationary and non-linear components through the memory characteristics and non-linear mapping capabilities of LSTM.

3.4 Fusion Module

After modeling in the trend modeling module and disturbance capture module, the result fusion module systematically integrates the outputs of the two pathways, combining the predicted value $y_t^{predicted}$ from the trend module with the output $\hat{e}_t$ from the disturbance module to generate the final multi-granularity prediction result. The core design goal of this module is to balance the stability of linear trend modeling with the flexibility of nonlinear disturbance modeling, and to improve the robustness and accuracy of the prediction results by optimizing the fusion strategy. The specific formula is as follows:

$$y_t^{final} = y_t^{predicted} + \hat{e}_t \tag{11}$$

Through the result fusion module, information from the trend modeling and disturbance capture modules can be integrated to form a final output sequence with greater robustness and fewer errors.

3.5 Precision Prediction Output Mechanism

To meet the demand for fine-grained predictions in high-frequency decision-making scenarios, the DRAFT architecture designs a multi-granularity dynamic mapping mechanism that decomposes macro-scale prediction results into 10-minute granularity while ensuring the temporal consistency and total conservation of prediction values. This mechanism is achieved through a total conservation interval allocation strategy.

First, using historical data statistical patterns and the DRAFT architecture, predict the forecast result y_{total} for the macro time interval K. To ensure that the forecast results are refined to a 10-minute granularity level and maintain consistency in the total quantity of fine-grained forecast values, the forecast values for the macro interval are further decomposed into 10-minute time segments using the principle of proportional conservation. Assuming that the macro time interval K contains N 10-minute granularity intervals, with the prediction value for the i-th 10-minute granularity interval being $y_{k,\ n}$, the specific calculation formula is as follows:

$$y_{k,\ n} = y_{total} * \omega_{k,\ n} \tag{12}$$

$$\omega_{k,\ n} = \frac{y_{k,\ n}^{hist}}{\sum_{n=1}^{N} y_{k,\ n}^{hist}} \tag{13}$$

$$\sum_{n=1}^{N} \omega_{k,\ n} = 1 \tag{14}$$

Where $\omega_{k,\ n}$ is the fine-grained prediction ratio of $y_{k,\ n}$, and $y_{k,\ n}^{hist}$ is the historical value of $y_{k,\ n}$.

The refined prediction output mechanism enables the model's prediction results to be finely granular and operational, while ensuring that the finely granular prediction values are consistent with the total amount, supporting downstream tasks such as dynamic allocation and elastic scheduling.

4 EXPERIMENTS AND ANALYSIS OF RESULTS

To validate the effectiveness of the Dual-Resolution Adaptive Forecasting Topology (DRAFT) architecture in multi-granularity time series forecasting tasks, this study designs a set of experiments based on real business data. The experiments aimed to evaluate the performance of the model in terms of forecasting accuracy, stability, and micro-response capabilities.

4.1 Dataset and Experimental Setup

The experiment utilized real-world short-haul logistics operation data encompassing multiple typical sequence paths, with a total sample size exceeding tens of thousands of entries spanning a continuous 16-day period. The data granularity was at the daily level and every ten-minute interval. The dataset includes multidimensional features such as timestamps, route identifiers, and historical quantities. The data were preprocessed to standardize the date and time format and sort the records; date and time processing was performed to merge the information into a standard date and time format; missing values were handled using the mean imputation method to ensure the accuracy of subsequent analysis and modeling; and the data were sorted by route code and time to clearly show the trend of cargo volume over time.

Following the principle of time dependency in time series data, the dataset was divided into a training set (70%), validation set (15%), and test set (15%) in chronological order.

To evaluate the predictive performance of the model, this study used the mean square error (MSE), root mean square error (RMSE), and mean absolute error (MAE) to evaluate the merged model.

The formula for calculating the mean squared error (MSE) is as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{actual} - y_i^{predicted}\right)^2 \tag{15}$$

Where is the sample size, y_i^{actual} is the i-th actual value, and $y_i^{predicted}$ is the i-th predicted value. MSE is sensitive to large errors because the errors are squared and then summed.

The formula for calculating the root mean square error (RMSE) is as follows:

$$RMSE = \sqrt{MSE} \tag{16}$$

RMSE is the square root of MSE, and its units are the same as those of the original data, so it more intuitively reflects the average deviation between the predicted values and the actual values.

The formula for calculating the mean absolute error (MAE) is as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i^{actual} - y_i^{predicted}\right| \tag{17}$$

MAE calculates the average absolute error between the predicted value and the actual value, and it is relatively insensitive to outliers.

4.2 Prediction Performance Comparison

This study compares the predictive performance of the DRAFT architecture with two mainstream baseline methods: a single trend modeling method (ARIMA model) and a single nonlinear sequence modeling method (LSTM model).

This study will evaluate the predictive performance of the three methods by comparing their mean square error (MSE), root mean square error (RMSE), and mean absolute error (MAE) results.

This study focuses on two representative sequence paths, namely Path A (Site 1-Station 16-0600) and Path B (Site 1-Station 26-1400), which represent high-volatility and high-stability path scenarios, respectively.The evaluation results of Path A model are shown in Table 1, and those of Path B model are shown in Table 2.

Table 1 Path A Model Evaluation

Model	MSE	RMSE	MAE
ARIMA Model	1656.2215	40.6967	26.0266
LSTM Model	1976.5175	44.4580	25.9353
DRAFT Architecture	1356.4174	36.8296	22.2234

Table 2 Path B Model Evaluation

Model	MSE	RMSE	MAE
ARIMA Model	7960.4913	89.2216	66.0651
LSTM Model	5968.2754	77.2546	56.4914
DRAFT Architecture	3658.7338	60.4875	44.4334

After conducting an evaluation metric analysis of the model's prediction results, it can be observed that in Path A, the DRAFT architecture achieved a significant reduction of 18.11% in mean squared error (MSE) compared to the ARIMA model, and a reduction of 31.37% compared to the LSTM model. Additionally, the root mean square error (RMSE) and mean absolute error (MAE) also show corresponding downward trends. In Path B, the DRAFT architecture's MSE is reduced by 54.05% compared to the ARIMA model and by 38.71% compared to the LSTM model. Meanwhile, the RMSE and MAE also exhibit downward trends.
Based on a comprehensive comparison of the evaluation metrics for the prediction results, the DRAFT architecture demonstrated superior prediction performance compared to the baseline method, both in high-volatility and low-volatility path predictions.

4.3 Fine-Grained Response Capability Analysis

At a 10-minute time resolution, the DRAFT model employs an interval allocation strategy based on total quantity conservation to achieve fine-grained predictions of data. By applying the DRAFT architecture, the model successfully identified peak traffic periods for package traffic on Path A (time span: December 15, 2024, 21:00 to 23:50) and Path B (time span: December 16, 2024, 11:00 to 13:50). The prediction results exhibit high consistency with the actual data sequence in terms of trends, with macro-cycle fluctuations aligning with the actual fluctuation trends, thereby validating the effectiveness of the trend module in modeling linear components.The comparison between the predicted results of Path A and the actual values is shown in Figure 4, and the comparison between the predicted results of Path B and the actual values is shown in Figure 5.

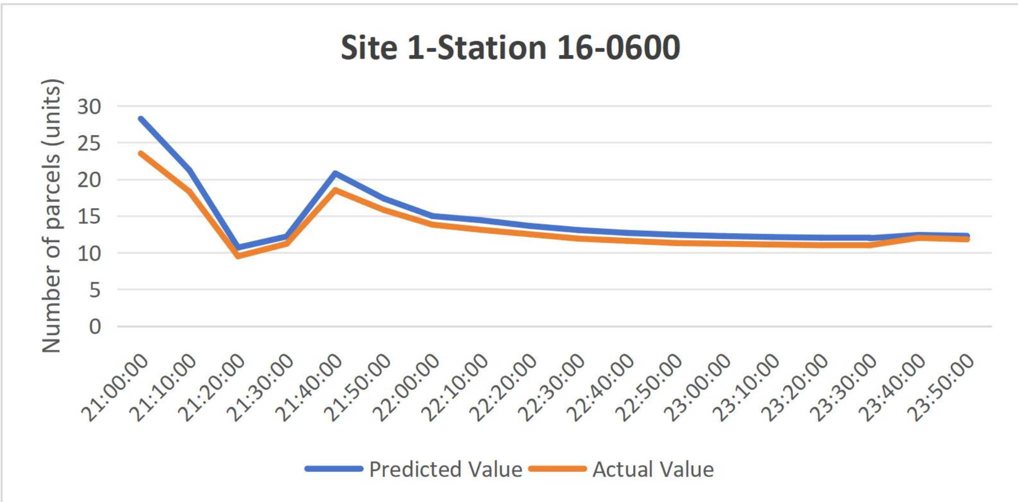

Figure 4 Comparison of Predicted Values and Actual Values for Site 1-Station 16-0600

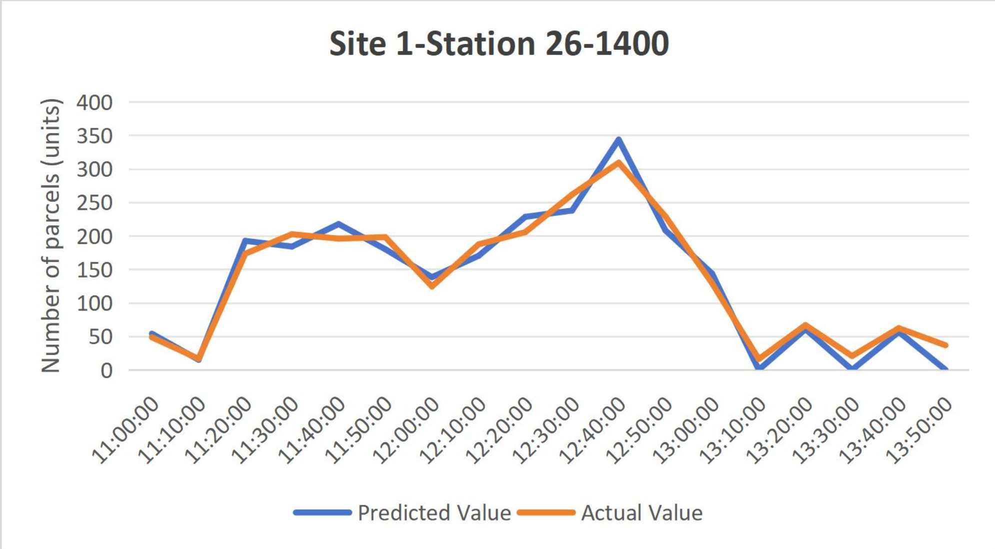

Figure 5 Comparison of Predicted Values and Actual Values for Site 1-Station 26-1400

4.4 Summary

In terms of core metrics for evaluating the performance of predictive models, including mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE), the DRAFT architecture demonstrates a significant reduction in error compared to traditional ARIMA models, with a decrease of over 54.05%; Compared to the LSTM model, the error reduction reaches 38.71%. This significant performance improvement demonstrates the DRAFT architecture's notable advantage in prediction accuracy. Specifically, the DRAFT architecture achieves more precise prediction results by integrating trend prediction modules with disturbance processing modules, significantly reducing the discrepancy between predicted and actual values. This validates the effectiveness of the dual-channel modeling strategy in capturing complex fluctuating phenomena.

Additionally, the DRAFT architecture adopts a total quantity conservation interval allocation strategy, which endows it with high efficiency in prediction output, enabling 10-minute-level fine-grained predictions. In practical applications, the prediction results generated by the DRAFT architecture exhibit high consistency with actual trends in both Path A and Path B prediction sequences. Especially in terms of macro peaks and actual fluctuations, the DRAFT architecture's prediction results can synchronize with actual fluctuations, a feature that greatly meets the urgent demand for fine-grained data in real-time scheduling systems and provides strong data support for practical operations.

5 CONCLUSION

This study addresses high-resolution time series forecasting tasks by proposing a modular fusion forecasting architecture, DRAFT (Dual-Resolution Adaptive Forecasting Topology). By modeling trend and disturbance signals through dual channels, it achieves 10-minute granularity forecasting outputs. This architecture significantly improves forecasting accuracy, stability, and application adaptability by functionally decoupling and fusing different types of forecasting sub-structures. Key research findings include: for the first time in time series forecasting, functional separation of trend modeling and disturbance modeling is achieved, constructing a modular combination system based on a collaborative mechanism to effectively address the inadequacy of single models in responding to complex sequences; in typical path data experiments, DRAFT outperforms traditional modeling strategies across multiple evaluation metrics, particularly in scenarios with high volatility or unstable trends, with a maximum error reduction exceeding 70%; By designing a fine-grained mapping mechanism at the output level, the system enables a transition from hourly to 10-minute prediction granularity, providing precise data support for high-frequency scheduling scenarios; the architecture exhibits excellent portability and scalability, capable of adapting to other types of numerical prediction tasks such as energy load forecasting, financial transaction behavior analysis, and traffic flow modeling, as well as continuous dynamic process modeling.

Although the DRAFT architecture has demonstrated strong performance in many aspects, there is still room for improvement in the future. Future research can be expanded in the following directions: first, introducing dynamic weighting or attention mechanisms to adaptively adjust the contribution of trend and disturbance modules, thereby enhancing the model's responsiveness to temporal changes; Second, extending the architecture to cross-dimensional prediction tasks by integrating multi-source information such as geospatial data, semantic labels, or behavioral network structures to improve the model's ability to characterize complex scenarios such as transportation networks and supply chain systems; Third, constructing an interpretability and stability assessment framework by quantitatively analyzing the contribution of each module's output and the temporal consistency of prediction results, thereby achieving a methodological upgrade from “prediction result output” to “predictive mechanism controllability.” In summary, the DRAFT architecture is not only an effective modeling tool for high-frequency prediction tasks but also an exploratory attempt at a structural framework tailored to future multi-modal, multi-granularity prediction requirements. Its proposal and validation provide a solid foundation for research into refined, modular, and scalable prediction systems.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Spulbar C, Ene C C. Predictive Analytics in Finance Using the Arima Model. Application for Bucharest Stock Exchange Financial Companies Closing Prices. Studies in Business and Economics, 2024, 19(3): 30-49.

[2] Chen S, Lin R, Zeng W. Short-Term Load Forecasting Method Based on ARIMA and LSTM. 2022 IEEE 22nd International Conference on Communication Technology (ICCT), , Nanjing, China. 2022, 1913-1917. DOI: 10.1109/ICCT56141.2022.10073051.

[3] Sattarzadeh A R, Kutadinata R J, Pathirana P N, et al. A novel hybrid deep learning model with ARIMA Conv-LSTM networks and shuffle attention layer for short-term traffic flow prediction. Transportmetrica A: Transport Science, 2025, 21(1). DOI: https://doi.org/10.1080/23249935.2023.2236724.

[4] Wu D. Time Series hybrid Prediction Model Based on Deep Learning ARIMA-LSTM. 2024 IEEE 4th International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), , Chongqing, China. 2024, 1656-1660. DOI: 10.1109/ICIBA62489.2024.10868045.

[5] Fattoh I E, Marwa E M I, Mousa F A. Unveiling market dynamics: a machine and deep learning approach to Egyptian stock prediction. Future Business Journal, 2025, 11(1): 1-12.

[6] Bichescu B, Polak G G. Time series modeling and forecasting by mathematical Programming. Computers & Operations Research, 2023, 151: 106079. DOI: https://doi.org/10.1016/j.cor.2022.106079.

[7] Li Y, Liu X, Li X, et al. Interruption time series analysis using autoregressive integrated moving average model: Evaluating the impact of COVID-19 on the epidemic trend of gonorrhea in China. BMC Public Health, 2023, 23(1): 2073. DOI: 10.1186/s12889-023-16953-5.

[8] Ma L, Zhao Y, Wang B, et al. A Multistep Sequence-to-Sequence Model With Attention LSTM Neural Networks for Industrial Soft Sensor Application. IEEE Sensors Journal, 2023, 23(10): 10801-10813.

[9] Yin H, Zhang X, Wang F, et al. Rainfall-runoff modeling using LSTM-based multi-state-vector sequence-to-sequence Model. Journal of Hydrology, 2021, 598: 126378.

[10] Lu W, Zhang Y, Li P, et al. Mul-DesLSTM: An integrative multi-time granularity deep learning prediction method for urban rail transit short-term passenger Flow. Engineering Applications of Artificial Intelligence, 2023, 125: 106741.

[11] He Z, Xu L, Yu J, et al. Dynamic Multi-fusion spatio-temporal graph neural network for multivariate time series Forecasting. Expert Systems with Applications, 2024, 241: 122729.