Integrating CEEMDAN-EWT and Hybrid Neural Network for Traffic Flow Forecasting

Yang Xu, Junyan Hu, Cheng Yao, Xiaorui Hu, Hexin Peng, Hongyu Long*

***Abstract*—To address the prediction challenges posed by the pronounced nonlinear characteristics and susceptibility to external interference in urban traffic flow data, this study proposes a hybrid forecasting framework integrating multimodal signal analysis and intelligent optimization. First, the raw traffic flow data are decomposed into multiscale components using Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN), followed by dynamic threshold-based Empirical Wavelet Transform (EWT) to adaptively filter high-frequency noise and enhance discriminative features. Second, a multidimensional feature space is constructed by integrating multisource spatiotemporal features, including meteorological factors and temporal attributes, while the Sparrow Search Algorithm (SSA) is employed to globally optimize the hidden layer dimensions of the Bidirectional Gated Recurrent Unit (BiGRU) network and the parameter configurations of its attention mechanism. Finally, XGBoost is utilized to synergistically integrate multimodal features and deep learning outputs for prediction refinement. Experimental results demonstrate that the proposed framework achieves near-optimal R^2 values while continuously reducing error metrics (MSE, MAE, RMSE, and MAPE), significantly outperforming standalone BiGRU, XGBoost, and conventional hybrid models. Robustness and generalizability are further validated on a second dataset with distinct spatiotemporal characteristics, demonstrating consistent performance. Systematic ablation studies confirm that each module contributes substantially to accuracy improvement, verifying the comprehensive design of the proposed framework.**

***Index Terms*—Traffic flow forecasting, Multi-modal decomposition, Hybrid neural network, Parameter optimization**

Manuscript received March 18, 2025; revised June 23, 2025.

This work was supported by the Chongqing Science and Technology Innovation and Application Development Special Project: Research and Application of Key Technologies for Electric Vehicle-Grid Interaction Integration (Project No. CSIB2024TIAD-KPX0089).

Yang Xu is a professor of Chongqing Automotive Electronics and Embedded Engineering Research Center, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: xuyang@cqupt.edu.cn).

Junyan Hu is a postgraduate student of Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: junyanhu01@163.com).

Cheng Yao is a senior engineer of Chongqing Public Transport Group Bayi Site Management Company, Chongqing 400020, China (e-mail: yaocheng_cq@163.com).

Xiaorui Hu is a professor level senior engineer of Marketing Service Center, State Grid Chongqing Electric Power Company, Chongqing 401123, China (e-mail: xiaorui4832@sina.com).

Hexin Peng is a postgraduate student of Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: penghexin715@163.com).

Hongyu Long is a professor level senior engineer of Chongqing Key Laboratory of New Energy Cyber-Physical Systems, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: longhongyu20@163.com).

I. Introduction

DRIVEN by urbanization and motorization, the rapid increase in motor vehicle ownership has led to increasingly severe traffic congestion, posing significant obstacles to socio-economic development. As an effective approach to mitigate traffic congestion, Intelligent Transportation Systems (ITS) prioritize accurate and timely traffic flow prediction as one of their core functionalities [1]. However, the complexity of traffic flow data—including its nonlinearity and volatility—coupled with external factors such as weather variations, renders short-term traffic flow prediction a highly challenging task. Traffic prediction is critical for effective traffic planning, management, and control. It enables urban travelers to select optimal routes, improving travel efficiency, while providing traffic authorities with a scientific basis to preemptively regulate congestion and implement vehicle restrictions. Over the past decade, with advancements in ITS, traffic flow prediction has garnered growing attention in both research and practical applications.

Current approaches to traffic flow forecasting are mainly divided into three key categories: traditional parametric techniques, conventional machine learning methods, and deep learning frameworks.

Traditional parametric techniques encompass models such as the Autoregressive Integrated Moving Average (ARIMA) [2], Exponential Smoothing [3], and Kalman Filtering [4], Multiple Regression [5], and Grey System Theory [6]. Among these, ARIMA and its variants [7-10] are suitable for forecasting stationary time series, while exponential smoothing excels in short-term predictions. The Kalman Filter demonstrates strong performance in dynamic scenarios, multiple regression models linear relationships between variables, and grey models are tailored for systems with incomplete information. However, these parametric approaches are often constrained by predefined assumptions, limiting their ability to capture the intricate dynamics of traffic flow. To address these limitations, researchers have developed hybrid models that integrate parametric methods with artificial intelligence (AI) techniques [11]. These hybrid frameworks exhibit greater flexibility in structural design and parameter tuning, enabling more effective handling of complex patterns and uncertainties inherent in traffic flow prediction.

Compared to classical parametric methods, traditional machine learning approaches are more adept at handling complex nonlinear problems. Support Vector Machine (SVM) [12][13] and Support Vector Regression (SVR) [14] enhance prediction accuracy by leveraging kernel functions to transform low-dimensional, linearly inseparable traffic data into high-dimensional, linearly separable representations. A novel hybrid model, denoted as C-WSVM, was proposed for short-term traffic speed prediction, which enhances the performance of Support Vector Machines (SVM) by incorporating wavelet functions and phase space reconstruction theory [15]. This methodology synergistically combines advanced signal processing and nonlinear dynamics to achieve superior predictive accuracy and robustness in complex traffic environments. Empirical validation using real-world data demonstrated the model's accuracy and practicality. Similarly, a short-term traffic flow prediction model was developed using Support Vector Regression (SVR) combined with a Genetic Algorithm (GA) for parameter optimization [16]. The feasibility and effectiveness of this model were confirmed through traffic flow data from the Jiangxi Provincial Department of Transportation. Urban traffic flow prediction framework integrating traffic, weather, calendar data, and Twitter information was proposed [17]. By employing SVM and SVR, their study empirically validated that such multimodal integration, including social media data form Twitter, significantly improves traffic flow prediction accuracy.

With the advancement of neural networks, an increasing number of scholars [18-21] have applied deep learning models to traffic flow prediction. These models excel at extracting hierarchical representations of features, enabling high-dimensional space modeling and parameter extraction. Variants of recurrent neural networks, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), are particularly suited for sequential data analysis, capturing long-term dependencies in time series. An innovative short-term traffic prediction model based on LSTM networks was proposed, enhancing prediction accuracy by integrating spatiotemporal correlations within traffic systems [22]. Comparative experiments with other models validated its superior performance. Similarly, LI [23] partitioned raw traffic flow sequences into multiple intervals to construct adaptive traffic flow series. By employing a GRU algorithm enhanced with data augmentation and a bidirectional GRU with adaptive mechanisms, their approach significantly improved prediction accuracy. As models evolve, hybrid deep learning frameworks [24-26] have gained traction, combining the strengths of individual models to address different aspects of data, thereby enhancing prediction accuracy and robustness. For example, literature [27] proposed a hybrid deep learning model that integrates an attention mechanism-enhanced Conv-LSTM network with Bi-LSTM. This model effectively captures spatial, short-term, and long-term temporal features of traffic flow, significantly enhancing the accuracy of short-term traffic predictions.

Short-term traffic flow data, often contaminated by noise during acquisition, faces challenges in reliability and accuracy due to spatiotemporal uncertainties and complexities in traffic systems. To address this, researchers have developed diverse data denoising techniques to enhance data quality prior to prediction, thereby improving precision and credibility.

Wavelet Transform (WT) has been widely adopted for noise reduction in traffic flow prediction. Wavelet-Kalman filter model was utilized to reduce noise in short-term traffic volume forecasting, combined with deep learning methods to validate the efficacy of wavelet-based processing [28]. KASHI [29] proposed a hybrid WT-ANN model, integrating wavelet denoising and artificial neural networks (ANN) to predict traffic flow 5 – 35 minutes ahead. Their results demonstrated that noise removal significantly enhanced prediction accuracy. Literature [30] proposed a CEEMDAN-XGBoost hybrid model, combining CEEMDAN and extreme gradient boosting, to achieve high-precision and stable lane-level traffic flow prediction, outperforming state-of-the-art methods such as ANN and LSTM. Literature [31] proposed a novel adaptive framework for traffic flow sequence analysis, utilizing CEEMDAN to decompose raw traffic flow time series into multiple adaptive subseries based on dataset characteristics, showcasing its superiority in capturing dynamic traffic flow variations.

This study addresses the complexity of urban traffic flow prediction by proposing an innovative hybrid neural network model based on multi-modal decomposition. To tackle the nonlinear and dynamic nature of traffic data, the model integrates CEEMDAN, EWT, SSA, an attention mechanism-enhanced BiGRU, and XGBoost to achieve precise traffic flow forecasting. The key contributions are as follows:

1) Multi-modal decomposition and denoising: The CEEMDAN algorithm is first applied to decompose traffic flow data into Intrinsic Mode Functions (IMFs) while removing noise. High-frequency IMFs are further decomposed via EWT to reduce residual noise and extract critical features, establishing a cleaner data foundation for prediction.
2) Attention-based BiGRU and XGBoost refinement: A BiGRU model enhanced with an attention mechanism is developed to capture bidirectional temporal dependencies and dynamically focus on key information points. XGBoost is then employed to refine predictions and improve generalization across diverse traffic scenarios.
3) Hyperparameter optimization via SSA: The SSA algorithm optimizes BiGRU hyperparameters (e.g., hidden layers, neuron counts), enhancing the model's adaptability to dynamic traffic patterns.
4) Multimodal feature integration: The model incorporates traffic, weather, and calendar features, enabling a holistic consideration of factors influencing traffic flow and significantly boosting prediction accuracy.
5) Comprehensive evaluation framework: A performance assessment system based on R^2, MSE, MAE, RMSE, and MAPE is established to rigorously validate the model's accuracy and reliability.

The paper is organized as follows: Section II details the theoretical foundations and mathematical frameworks of CEEMDAN, EWT, BiGRU, and XGBoost. Section III elaborates on the application of SSA for BiGRU hyperparameter tuning and the structural design of the hybrid model. Section IV presents empirical experiments, ablation studies to validate the importance of each module, and performance evaluations to demonstrate the efficacy of the proposed approach. Finally, the conclusion summarizes key findings and discusses future research directions.

II. METHOD

In this section, we comprehensively elucidate the core principles of the algorithms and methodologies employed in this study. Specifically, this research delves into the implementation mechanisms of CEEMDAN, EWT, BiGRU, and XGBoost. These methods not only constitute critical components of the proposed hybrid forecasting model but also play a pivotal role in ensuring the accuracy and stability of the final prediction results.

A. Complete Ensemble Empirical Mode Decomposition with Adaptive Noise

CEEMDAN algorithm is an enhanced variant of the Ensemble Empirical Mode Decomposition (EEMD). It aims to achieve precise reconstruction of original signals and improved spectral separation of modes while reducing computational costs. CEEMDAN is specifically designed for processing nonlinear and non-stationary signals. By adaptively introducing noise and iterating multiple times, it effectively mitigates the mode mixing issue inherent in traditional Empirical Mode Decomposition (EMD), thereby enhancing decomposition accuracy and stability.

The core principle of CEEMDAN lies in incrementally adding white noise to the IMFs obtained from EMD decomposition during each iteration. This stepwise noise addition significantly reduces the impact of white noise on decomposition results, improving both precision and robustness. Unlike EEMD and CEEMD, which perform global averaging across all IMF components, CEEMDAN applies averaging immediately after each IMF is generated. This ensures that noise effects are minimized at every decomposition stage. Subsequently, the residual component is processed iteratively, effectively addressing the transfer of white noise from high-frequency to low-frequency bands.

The key steps for implementing the CEEMDAN algorithm are outlined below, with Algorithm 1 providing the pseudocode:

Step 1: Initialization and Noise Addition. For the original signal *Y(n)*, multiple realizations of Gaussian white noise $\epsilon_k(n)$ (with zero mean and variance σ^2) are added to generate a noisy signal sequence. Let *K* denote the number of noise realizations. For each realization *k=1, 2, ..., K*, the noisy signal is defined as $Y_k(n) = Y(n) + \epsilon_k(n)$.

Step 2: EMD Decomposition. Apply EMD to each noisy signal $Y_k(n)$, decomposing it into IMF components and a residual R_k:

$$Y_k(n) = IMF_{k1}(n) + ... + IMF_{kM}(n) + R_k(n) \tag{1}$$

Step 3: Compute IMF Averages. For each mode *m=1, 2, ..., M*, compute the average across all *K* realizations of IMF_m to obtain the CEEMDAN-derived IMF_m :

$$IMF_m(n) = \frac{1}{K}\sum_{k=1}^{K} IMF_m^k(n) = \overline{IMF_m}(n) \tag{2}$$

Step 4: Residual Calculation and Iteration. Subtract the averaged IMF components from the original signal to generate the residual sequence (3):

$$R(n) = Y(n) - \sum_{m=1}^{M} IMF_m(n) \tag{3}$$

The residual sequence *R(n)* is then augmented with Gaussian white noise, and Steps 2–3 are repeated to derive the next IMF component.

Step 5: Iterative Decomposition. Repeat Steps 2–4 for each new residual sequence until the residual contains fewer than two extrema (local maxima or minima) or meets predefined stopping criteria.

Step 6: Stopping Criteria. Terminate the algorithm when the residual magnitude becomes negligible, no new IMFs can be extracted, or a preset maximum iteration count is reached.

Step 7: Signal Reconstruction. The original signal is reconstructed by summing all IMFs and the final residual:

$$Y(n) = \sum_{m=1}^{M} IMF_m(n) + R(n) \tag{4}$$

Through these steps, CEEMDAN effectively decomposes nonlinear and non-stationary signals into a series of IMFs, each representing distinct frequency modes. The pseudo-code of CEEMDAN is illustrated in TABLE I.

TABLE I
PSEUDO-CODE OF THE CEEMDAN ALGORITHM

```
Algorithm 1 The Realization of the CEEMDAN Approach
Input: Y(n)-The raw time-series data
       K- The number of realizations
Output: $IMF_m$ , m=1, 2, ..., M and R(n)

for k=1 to K do
    $Y_k(n) \leftarrow Y(n) + \epsilon_k(n)$
end
for k=1 to K do
    $[IMF_{k1}(n), IMF_{k2}(n), \ldots, IMF_{kM}(n), R_k(n)] \leftarrow EMD(Y_k(n))$
end
for k=1 to K do
    $IMF_m(n) \leftarrow \left(\sum_{k=1}^{K} IMF_m^k(n)\right) / K$
end
$R(n) \leftarrow Y(n) - \sum_{m=1}^{M} IMF_m(n)$
while Num (R(n)) ≥ 2 do
    for k=1 to K do
        $Y_k(n) \leftarrow R(n) + \epsilon_k(n)$
        $[IMF_{k1}(n), IMF_{k2}(n), \ldots, IMF_{kM}(n), R_k(n)] \leftarrow EMD(Y_k(n))$
    end
    for k=1 to K do
        $IMF_m(n) \leftarrow \left(\sum_{k=1}^{K} IMF_m^k(n)\right) / K$
    end
    $R(n) \leftarrow Y(n) - \sum_{m=1}^{M} IMF_m(n)$
end
$r(n) \leftarrow R(n)$
```

B. Empirical Wavelet Transform

The EWT proposed by Gilles in 2013, combines the locality of wavelet transforms with the adaptability of Empirical Mode Decomposition to address limitations of traditional wavelet methods in analyzing nonlinear and non-stationary signals. EWT designs adaptive wavelet filter banks to extract meaningful subbands from non-stationary time series.

EWT Implementation Steps:

1. Extend the original signal to facilitate the Fourier transform.

2. Compute the Fourier transform of the input signal, converting the time-domain signal to the frequency domain.

3. Partition the Fourier spectrum into *N* continuous

segments. Determine the boundaries by identifying local maxima in the spectrum and sorting them in descending order. If the number of local maxima *M* satisfies *M≥N*, retain the first *N−1* maxima; if *M<N*, retain all maxima and adjust *N* accordingly. Finally, define the midpoint frequency *ωn* between two adjacent local maxima as the boundary.

4.Based on the local maxima identified in Step 3, determine the segmentation boundaries and partition the spectrum.

5.Construct an appropriate wavelet filter bank, which will be used to decompose the signal.

6.Decompose the signal using the constructed wavelet filter bank to extract subbands. This step breaks the signal into a finite number of modes, each corresponding to a subband of the spectrum.

7.Decompose the signal into several uncorrelated filtering modes and a residual. These modes represent different intrinsic patterns of the signal, while the residual contains components not explained by the modes.

C. Bidirectional Gated Recurrent Unit-Attention

The BiGRU-Attention architecture addresses sequential modeling challenges through integrated gating mechanisms and selective feature weighting. This framework employs GRU cells - a streamlined variant of recurrent networks that mitigates gradient issues through dual gating control. Unlike LSTM's three-gate design, GRU utilizes an update gate Z_t and reset gate r_t to regulate information flow, the computational formulas (5) for GRU are as follows:

$$\begin{aligned} z_t &= \sigma(W_z \cdot [x_t, h_{t-1}] + b_z) \\ r_t &= \sigma(W_r \cdot [x_t, h_{t-1}] + b_r) \\ \tilde{h}_t &= \tanh(W \cdot [x_t, r_t \odot h_{t-1}] + b) \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \end{aligned} \tag{5}$$

in (5), σ denotes the sigmoid activation, ⊙ represents Hadamard product, and ***W*.** and ***b*.** are learnable parameters. The update gate modulates historical state retention, while the reset gate controls previous state contribution to candidate activation $\tilde{h}_t$.

BiGRU combines two unidirectional GRU models—to capture comprehensive temporal dependencies, bidirectional processing integrates complementary directional contexts. At each time step, the hidden state of the BiGRU is a combination of the hidden states from the forward and reverse GRUs, which can be either concatenated or weighted summed. The computational formula (6) for BiGRU is as follows:

$$h_t^{BiGRU} = \varnothing(h_t^{forward}, h_t^{backward}) \tag{6}$$

where, $h_t^{forward}$ and $h_t^{backward}$ encode forward/backward sequence contexts, fused via concatenation or weighted summation (Ø).

The introduction of the attention mechanism aims to simulate the selective focus of humans when processing information, enabling the model to identify and concentrate on the most important parts when handling large amounts of data. In sequence processing tasks, this means the model needs to recognize the input segments that are most influential to the current task and allocate more attention to them. The query (Q), key (K), and value (V) vectors are constructed from the outputs of the BiGRU. Typically, these vectors are derived through linear transformations of the hidden states of the BiGRU, formally defined as (7):

$$\begin{aligned} Q &= W_q H^{BiGRU} \\ K &= W_k H^{BiGRU} \\ V &= W_v H^{BiGRU} \end{aligned} \tag{7}$$

where, H^{BiGRU} represents the output of the BiGRU layer, and W_q, W_k and W_v are learnable weight matrices.

The similarity between the query vector and all key vectors is computed, followed by weight allocation using the softmax function, which can be expressed by the formula (8):

$$\begin{aligned} e_{t,j} &= v_a^T \tanh(Q_t W_k^T K_j) \\ \alpha_{t,j} &= \frac{\exp(e_{t,j})}{\sum_{k=1}^{T} \exp(e_{t,k})} \end{aligned} \tag{8}$$

The context vector is generated through weighted summation of the attention weights. Pass the context vector c_t to the output layer to make the final decision.

D. XGBoost

XGBoost follows the gradient boosting framework, which is an ensemble learning method that improves prediction performance by iteratively adding new tree models to correct the residuals or errors of the previous models. In each iteration, the newly added tree model focuses on the prediction errors of the previous model, thereby enhancing the overall predictive accuracy.

1) Objective Function

The objective function (9) synergistically combines predictive error minimization and structural regularization:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \Omega\left(f_t\right) \tag{9}$$

where, $l\left(y_i, \hat{y}_i^{(t)}\right)$ represents the loss function, y_i denotes the true value of the sample *i*, $\hat{y}_i^{(t)}$ denotes the predicted value of the sample *i* by the model, and $\Omega(f_t)$ represents the regularization term for the complexity of the *t*-th tree model.

2) Loss function

The loss function is used to measure the difference between the predicted values and the actual values. Common loss functions include squared loss and logarithmic loss. For the *i*-th sample, the loss function can be expressed as (10):

$$l(y_i, \hat{y}_i) = \left(y_i - \hat{y}_i\right)^2 \tag{10}$$

3) Regularization

The regularization term (11) is used to measure the complexity of the tree, preventing overfitting:

$$\Omega\left(f_t\right) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{11}$$

where, *T* is the leaf count, w_j represents leaf weights with *γ*, *λ* controlling model complexity.

4) Second-order Taylor expansion

XGBoost utilizes Taylor expansion to approximate the loss function as a second-order polynomial, facilitating optimization. The loss function is approximated via second-order Taylor expansion at iteration *t*:

$$l(y_i, \hat{y}_i) \approx l(y_i, \hat{y}_{i,}^{t-1}) + g_i(\hat{y}_i - \hat{y}_i^{t-1}) + \frac{1}{2} h_i (\hat{y}_i - \hat{y}_i^{t-1})^2 \tag{12}$$

in (12), $g_i = \partial_{\hat{y}} l \big|_{\hat{y}=\hat{y}^{(t-1)}}$ and $h_i = \partial^2_{\hat{y}} l \big|_{\hat{y}=\hat{y}^{(t-1)}}$ as gradient and Hessian components. Removing constants yields the optimized objective.

That is, the objective function can be simplified to (13):

$$\begin{aligned}\mathcal{L}^{(t)} &= \sum_{i=1}^{n}\left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t\left(x_i\right) + \frac{1}{2} h_i f_t^2\left(x_i\right)\right] + \Omega\left(f_t\right) \\ &= \sum_{i=1}^{n}\left[g_i f_t\left(x_i\right) + \frac{1}{2} h_i f_t^2\left(x_i\right)\right] + \Omega\left(f_t\right) + c\end{aligned} \qquad (13)$$

where, $\hat{y}_i^t = \hat{y}_i^{t-1} + f_t(x_i)$, Based on the principle of Boosting, it can be derived that: the prediction value of the sample *i* by the *t*-th tree = the prediction value of the previous *t-1* trees + the prediction value of the *t*-th tree, *c* is a constant term.

Leveraging Boosting's additive training principle, XGBoost reformulates the objective function via second-order Taylor approximation for loss optimization. Subsequent elimination of invariant terms yields the minimized expression for deriving optimal leaf weights (14):

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n}\left[g_i f_t\left(x_i\right) + \frac{1}{2} h_i f_t^2\left(x_i\right)\right] + \Omega\left(f_t\right) \qquad (14)$$

5) Leaf node weights

For a fixed tree structure *q* with leaf partition $I_j = \{i | q(x_i) = j\}$, optimal weights are derived as (15):

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n}\left[g_i w_j + \frac{1}{2} h_i w_j^2\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \qquad (15)$$

Then $G_j = \sum_{i \in I_j} g_i$ and $H_j = \sum_{i \in I_j} h_i$, let it be further simplified to (16):

$$\mathcal{L}^{(t)} = \sum_{j=1}^{T}\left[G_j w_j + \frac{1}{2}\left(H_j + \lambda\right) w_j^2\right] + \gamma T \qquad (16)$$

For a fixed tree structure, the optimal weight w_j for the *j*-th leaf node can be expressed as: $w_j = -\frac{G_j}{H_j + \lambda}$ .Based on this, the optimal value of the objective function can be derived (17) to yielding the structure score::

$$\mathcal{L}^{(t)*} = -\frac{1}{2}\sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T \qquad (17)$$

6) Splitting gain calculation

In practical applications, enumerating all potential tree structures is computationally infeasible due to complexity constraints. Therefore, XGBoost employs a greedy algorithm to construct the tree model by iteratively selecting the optimal split points, rather than exploring all possible tree structures. During the splitting process, a gain function (18) is used to evaluate the effectiveness of a split:

$$\text{Gain} = \frac{1}{2}\left(\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right) - \gamma \qquad (18)$$

where, G_L, G_R, H_L, H_R are the sums of the gradients and the second-order derivatives (Hessians) of the left child node, respectively, while G_L, G_R, H_L, H_R are those of the right child node. λ is the regularization parameter.

7) Construction of the tree structure

XGBoost constructs the tree structure by iteratively adding branches until a predefined maximum depth is reached or no further gain is achieved.

8) Prediction and Output

After the model training is completed, for a new input sample, XGBoost sequentially inputs the sample features into each decision tree and sums the predictions from all trees to obtain the final prediction output.

Through the above steps, the pseudo-code of the XGBoost is described in TABLE II. XGBoost effectively combines multiple decision trees and iteratively optimizes the model using gradient boosting, ultimately achieving superior predictive performance.

TABLE II
PSEUDO-CODE OF THE XGBOOST ALGORITHM

```
Algorithm 2 XGBoost Training Process
Input: X-Feature matrix; y-Labels; T_max-Max trees;
       η-Learning rate; λ,γ-Regularization; max_depth-Tree depth
Output: Trees-Ensemble of trained trees
# Initialize base prediction
$\hat{y}_i^{(t)}$ = base_prediction(y)
for t=1 to T_max
    $g_i \leftarrow \frac{\partial l(y_i, \hat{y}_i^{t-1})}{\partial \hat{y}_i^{t-1}}$ , $h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{t-1})}{(\partial \hat{y}_i^{t-1})^2}$
    function BuildTree(X, g, h, depth):
    if depth ≥ max_depth or Gain < γ:
        $w_j \leftarrow -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$
     return leaf node with wj
    else
  Gain $\leftarrow \frac{1}{2}\left( \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right) - \gamma$
    X_left={x_i|x_split_feature≤split_value}
    X_right={x_i|x_split_feature>split_value}
    left_child=BuildTree(X_left, g_left, h_left, depth+1)
    right_child=BuildTree(X_right, g_right, h_right, depth+1)
     return internal node with split rule
     $\hat{y}_i^{(t)} \leftarrow \hat{y}_i^{(t-1)} + \eta f_t(x_i)$
     Trees=Trees ∪ {f_t}
  end for
return Trees
```

III. Algorithms and Model Framework

This research employs the SSA to optimize pivotal hyperparameters within the traffic flow prediction framework. By mimicking sparrow swarm dynamics, SSA determines optimal weight distributions for the multimodal BiGRU-XGBoost architecture integrating traffic, meteorological, and temporal features. This optimization minimizes adaptive prediction errors through targeted adjustment of BiGRU-Attention structural parameters, including hidden layer quantities and neuronal configurations per layer. Consequently, the enhanced model exhibits superior capability in capturing both transient fluctuations and longitudinal trends in traffic flow dynamics.

A. Sparrow Search Algorithm

The SSA constitutes a swarm intelligence optimizer inspired by sparrows' foraging and collective movement patterns. Within this framework, the population segregates into explorers and followers. Explorers orchestrate food sourcing and guide collective foraging trajectories, while followers capitalize on resources located by explorers. Individuals engage in mutual behavioral monitoring, competing with high-yield conspecifics upon resource discovery to maximize nutritional intake efficiency. Predator detection triggers coordinated evasion responses.

The SSA prioritizes explorers with superior adaptation levels during search operations. Given their role in sustaining collective nutrition, explorers command broader foraging radii compared to followers. Nutritionally disadvantaged followers (low fitness values) exhibit area-shifting behavior to replenish energy reserves. Imminent threat perception prompts rapid relocation to secure foraging zones. Algorithmic implementation follows these core principles, and pseudocode in Algorithm 2:

First, population initialization commences with randomized positioning (19):

$$X=\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & \cdots & x_{n,d} \end{bmatrix} \tag{19}$$

where, the SSA initializes with n agents in d-dimensional space, represented by position matrix $X\in R^{n\times d}$. Each row vector x_i corresponds to an agent's coordinates. The fitness values of all sparrows are computed to form a fitness value vector F, as shown in the following equation (20):

$$F_X=\begin{bmatrix} f([x_{1,1} & x_{1,2} & \cdots & \cdots & x_{1,d}]) \\ f([x_{2,1} & x_{2,2} & \cdots & \cdots & x_{2,d}]) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f([x_{n,1} & x_{n,2} & \cdots & \cdots & x_{n,d}]) \end{bmatrix} \tag{20}$$

whose components F_i determine role assignment: high-fitness individuals become explorers (producers), while low-fitness agents serve as followers (scroungers). Explorers command expanded search domains due to their nutritional responsibility. Position updates for explorers follow dual modalities based on threat assessment. By iteratively updating the positions of discoverers using the following two formulas (21):

$$X_{i,j}^{t+1}=\begin{cases} X_{i,j}^{t}\cdot\exp\left(\dfrac{-i}{\alpha\cdot\text{iter}_{\max}}\right) & \text{if} R_2<ST \\ X_{i,j}^{t}+Q\cdot L & \text{if} R_2\geq ST \end{cases} \tag{21}$$

within the SSA framework, t denotes the current iteration index, while t_{max} specifies the maximum iterations. The position of sparrow i in dimension j at iteration t is denoted $X_{i,j}^{t}$. Randomness is modeled through $\alpha\in(0,1]$, with $R_2\in[0,1]$ representing alertness level and $ST\in[0.5,1.0]$ defining safety threshold. Environmental stochasticity is simulated by $Q\sim N(0,1)$, and $L=1_{1\times d}$ ensures dimensional integrity during updates. When $R_2<ST$ indicates that absence of predators enables explorers to conduct wide-range exploration. When $R_2\geq ST$ indicates that collective predator detection triggers immediate relocation to secure foraging zones.

Followers continuously surveil explorers, initiating competitive foraging upon resource discovery. Success grants immediate consumption; failure prompts continued search. Follower position updates (22) incorporate:

$$X_{ij}^{t+1}=\begin{cases} Q\cdot\exp\left(\dfrac{x_{\text{worst}}^{t}-x_{ij}^{t}}{i^2}\right) & \text{if } i>\dfrac{n}{2} \\ X_p^{t+1}+|X_{ij}^{t}-X_p^{t+1}|\cdot A^{+}\cdot L & \text{otherwise} \end{cases} \tag{22}$$

where, X_p denotes the optimal explorer position, while X_{worst} signifies the globally worst recorded position. Matrix $A\in\{-1,1\}^{1\times d}$ contains stochastically assigned elements, with its pseudo-inverse defined as $A^+=A^T(AA^T)^{-1}$. The condition $i>n/2$ identifies energetically deficient followers (low fitness) requiring relocation for energy replenishment.

When alertness R_2 surpasses safety threshold ST, emergency evasion is activated. This can be achieved using the following formula (23):

$$X_{ij}^{t+1}=\begin{cases} X_{\text{best}}^{t}+\beta\cdot|X_{ij}^{t}-X_{\text{best}}^{t}| & \text{if } f_i>f_g \\ X_{ij}^{t}+K\cdot\left(\dfrac{|X_{ij}^{t}-X_{\text{worst}}|}{(f_i-f_w)+\varepsilon}\right) & \text{if } f_i=f_g \end{cases} \tag{23}$$

here, the global optimum solution is denoted X_{best}, with $\beta\sim N(0,1)$ controlling step magnitude. Directional coefficient $K\in[-1,1]$ governs movement orientation, while f_i, f_g, and f_w represent current, global optimum, and worst fitness values respectively. Constant $\varepsilon<<1$ prevents division singularities.

When $f_i>f_g$ that peripheral agents face elevated predation vulnerability. When $f_i=f_g$ that core agents initiate cohesion behavior to mitigate predation risk. Termination occurs at $t=t_{\max}$, outputting $\boldsymbol{x}_{best}$ and f_g as the optimal solution, the pseudocode of the SSA is illustrated in TABLE III.

TABLE III
PSEUDOCODE OF THE SSA

Algorithm 3 The Realization of the SSA Approach

```
# Initialization Step
Randomly initialize the population of sparrows:
X = [X_1, X_2, ..., X_n]
Set parameters: $\alpha$, $Q$, $L$, $iter_{max}$, $ST$ (stopping criteria)
Calculate the fitness value $f(X_i)$ for each sparrow
# Iterative search process
while stopping criteria $ST$ is not met:
Compute the fitness value $f(X_i)$
Record the current best position $X_{best}$ and worst position $X_{worst}$
    # Leader (explorers) update their positions
        for $i$ in range($p$ * $n$):
            if r < ST:
                X[$i$] ← X[$i$] * exp(-$i$ / ($\alpha$ * $iter_{max}$))
            else:
                X[$i$] ← X[$i$] + $Q$ * $L$
        End
    Constrain X[$i$] within search boundaries
    # Followers (joiners) update their positions
        for $i$ in range($p$ * $n$, $n$):
            X[$i$] ← Q * exp(($X_{worst}$ - X[$i$]) / ($i$^2))
        End
    Constrain X[$i$] within search boundaries
    # Check fitness values and update best and worst positions
    Update $X_{best}$, $X_{worst}$
# Output the best solution
Output $X_{best}$, the best position found during the iterations
```

B. Measures Of Effectiveness

Prediction performance is evaluated using three metrics: R², MSE, MAE, RMSE, and MAPE. By comparing these parameters, we can more thoroughly assess the performance of the proposed model. The mathematical expressions for each evaluation metric are as follows (24), (25), (26), (27), (28):

$$R^2 = 1 - \frac{\sum_i (y(i) - y(i))^2}{\sum_i (\bar{y} - y(i))^2} \tag{24}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}[y(i) - y(i)]^2 \tag{25}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y(i) - y(i)\right| \tag{26}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[y(i) - y(i)\right]} \tag{27}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y(i) - y(i)}{y(i)}\right| \tag{28}$$

here, *y(i)* represents the measured traffic flow data at time *i*, $y(i)$ is the predicted traffic flow value, and *N* is the number of data points. Smaller values of MSE, MAE, RMSE, and MAPE indicate better performance, while an R² value closer to 1 indicates better performance. These metrics effectively and comprehensively illustrate the quality of the model's prediction results.

The following evaluation criteria can thoroughly and accurately demonstrate the predictive power of a model in terms of outcomes. Utilizing these assessment metrics, we will proceed to evaluate the model's performance in the subsequent section.

C. A Forecasting Model Based On CEEMDAN

This research proposes a cohesive computational architecture for traffic flow forecasting, integrating state-of-the-art methodologies to augment predictive precision and stability. Fig 2 delineates the framework's structural composition, comprising these critical components:

a) Data Preprocessing Module

Before model training, preprocessing the raw traffic data is a fundamental and critical step. This study employs the CEEMDAN algorithm to adaptively decompose the data, effectively removing noise and extracting the IMFs of traffic flow. This step provides clearer input features for the model, contributing to improved prediction accuracy. In the traffic flow prediction model, the processing of high-frequency IMFs is particularly crucial. High-frequency IMF components, such as IMF 1, IMF 2, and IMF 3, often exhibit high energy density in the spectrum, which may indicate the presence of high-frequency noise or interference. These high-frequency components may not contribute to the prediction task and could even reduce prediction accuracy. Therefore, further processing of these high-frequency IMF components is necessary. To extract meaningful signals and reduce noise, we apply EWT to deeply process the high-frequency IMF components. EWT decomposes high-frequency signals into more stable components, not only helping to remove high-frequency noise but also ensuring that the data input to the model is smoother and more conducive to capturing the intrinsic patterns of traffic flow. Through this approach, we can extract more accurate and predictive features from the raw data, thereby enhancing the overall performance of the traffic flow prediction model.

b) Feature Extraction Module

In this study, we comprehensively consider various factors influencing traffic flow, including time, speed (mph), holidays, temperature, traffic flow, and weather conditions. To ensure the model captures factors significantly impacting traffic flow, we quantify the importance of these factors using statistical methods such as correlation analysis. The results of this evaluation not only help the model identify key factors but also optimize the model's sensitivity to traffic flow changes by assigning appropriate weights to each factor. Subsequently, the feature extraction module utilizes the BiGRU-Attention model to perform deep feature extraction on the preprocessed data. By combining the bidirectional GRU structure and the attention mechanism, this model effectively captures dynamic changes and key information points in the time series. This integration enables the model to more precisely extract core features related to traffic flow, thereby improving prediction accuracy.

c) Parameter Optimization Module

To further enhance model performance, this study introduces the SSA to optimize the hyperparameters of the BiGRU-Attention model. The SSA algorithm intelligently searches the solution space by simulating the social behavior of sparrows, identifying the optimal number of hidden layers and neurons per layer.

d) Refinement Prediction Module

Building on the BiGRU-Attention model, the XGBoost model is introduced to refine the prediction results. XGBoost, with its excellent generalization ability and capability to handle nonlinear problems, adjusts the output of the BiGRU-Attention model to improve prediction accuracy. This step further enhances the model's predictive performance by refining the results. The architecture of the proposed BiGRU-Attention-XGBoost model is shown in Fig 2, which integrates bidirectional sequence learning, attention mechanisms, and gradient boosting.

e) Model Integration

The design of the entire model framework aims to achieve high-precision traffic flow prediction by integrating multiple advanced machine learning techniques. The synergistic collaboration of CEEMDAN denoising, SSA optimization, BiGRU-Attention feature extraction, and XGBoost refinement prediction enables the model to comprehensively capture the dynamic characteristics of traffic flow, improving the reliability and accuracy of predictions.

Overall, as illustrated in Fig 2, the proposed CEEMDAN-EWT-SSA-BiGRU-XGBoost architecture sequentially integrates adaptive signal decomposition, wavelet filtering, swarm intelligence optimization, and hybrid prediction modules, forming an end-to-end framework for traffic flow forecasting.

IV. EXPERIMENTS AND RESULTS ANALYSIS

A. *Dataset Description and Preprocessing*

This research employs a traffic flow from a mountainous city in southwestern China, spanning one month with a 5-minute resolution, yielding 8,904 data points. The dataset is split into training and test sets, with 80% allocated for training and 20% for testing, as depicted in Fig 3, to assess the model's performance on unseen data. It accurately captures traffic variations on key urban roads in the region. Prior to training, preprocessing steps were taken including outlier removal and missing value imputation to ensure data quality. To improve the model's ability to detect dynamic traffic patterns, temporal features like time, day, date, month, and notable events were incorporated, enhancing the model's capacity to model temporal relationships in traffic flow.

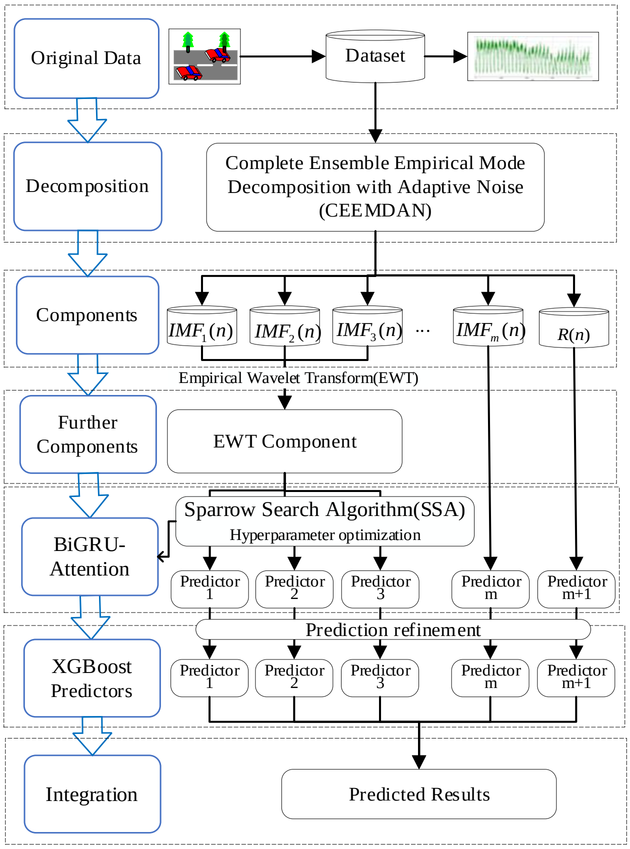


Fig 1. CEEMDAN-EWT-SSA-BiGRU-XGBoost Model Architecture

B. *Data Denoising*

First, we applied the CEEMDAN algorithm to adaptively decompose the original traffic data, effectively reducing noise levels and extracting key traffic flow features. The CEEMDAN algorithm decomposes the original data into several IMFs, which capture the main components of the data while removing or reducing the impact of noise. The CEEMDAN decomposition results, as illustrated in Fig 4, decomposes the original traffic flow data into multiple IMFs, revealing the primary components of the data. For IMF 1, IMF 2, and IMF 3 obtained from CEEMDAN decomposition, their high energy density in the spectrum suggests the presence of high-frequency noise or interference, necessitating further processing. We employed EWT to further decompose these high-frequency IMFs into components of different frequencies, identifying and separating high-frequency noise components from low-frequency trend components. The EWT decomposition results of three selected IMFs are shown in Fig 5, where (a)-(c) illustrate the frequency components of IMF 1, IMF 2, and IMF 3, respectively.

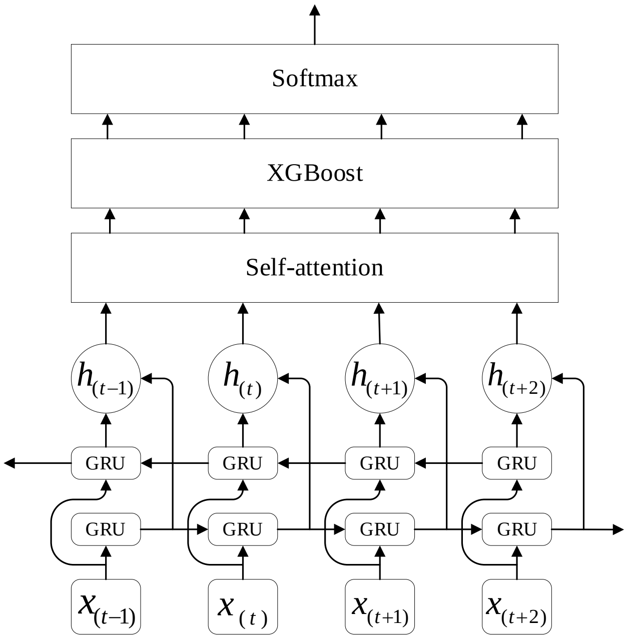


Fig 2. BiGRU-Attention-XGBoost Model Architecture Diagram

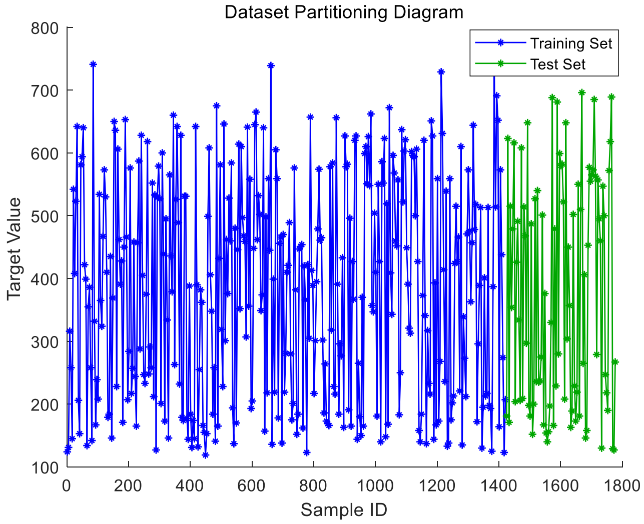


Fig 3. Dataset Partitioning Diagram

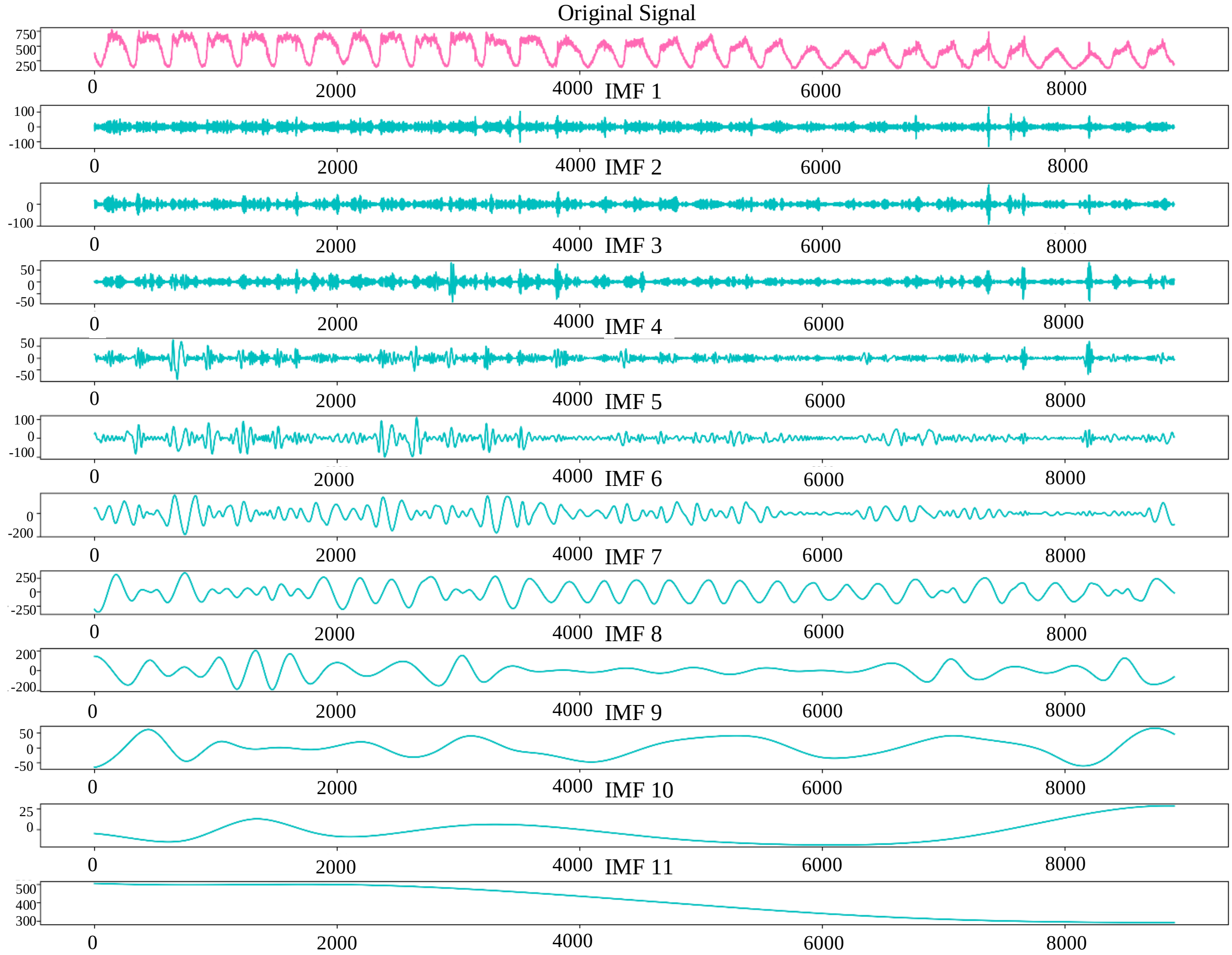
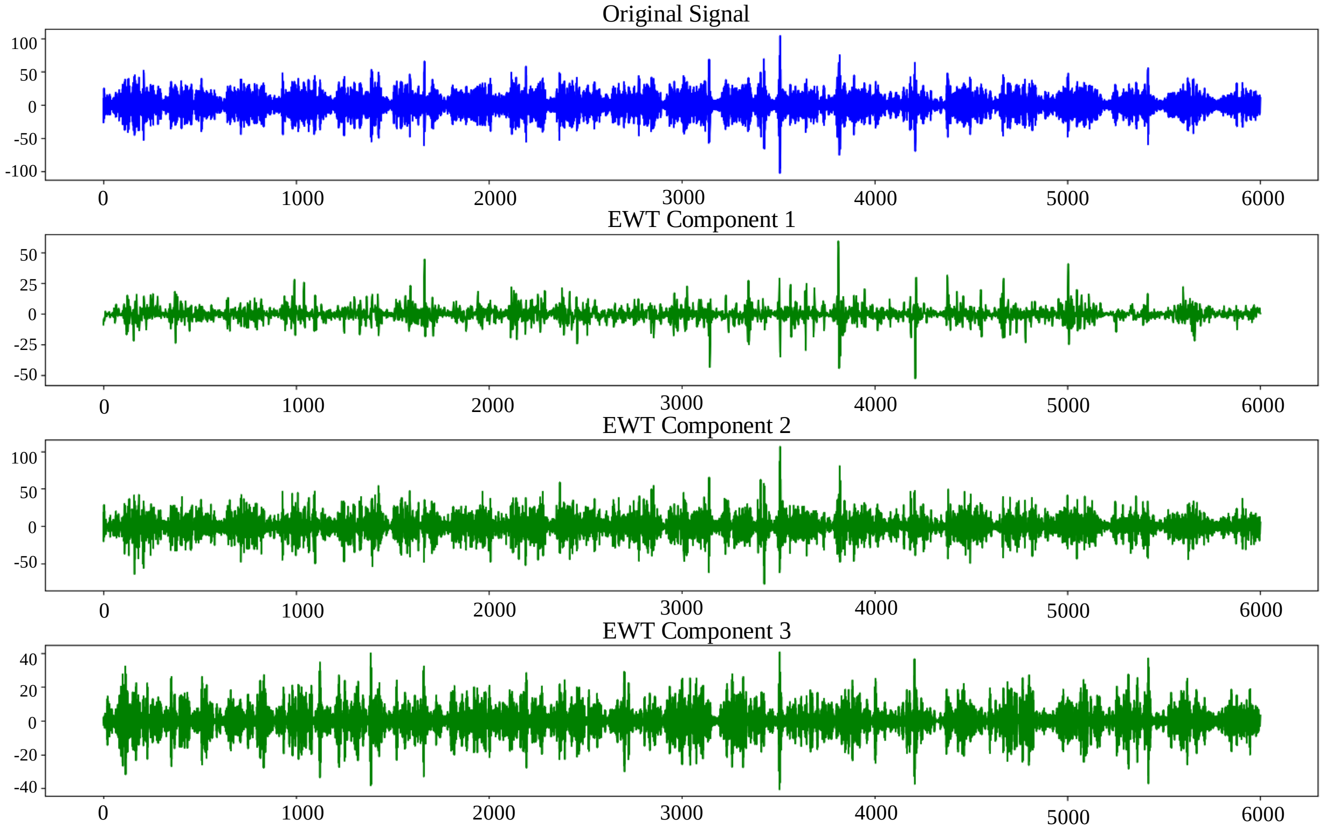
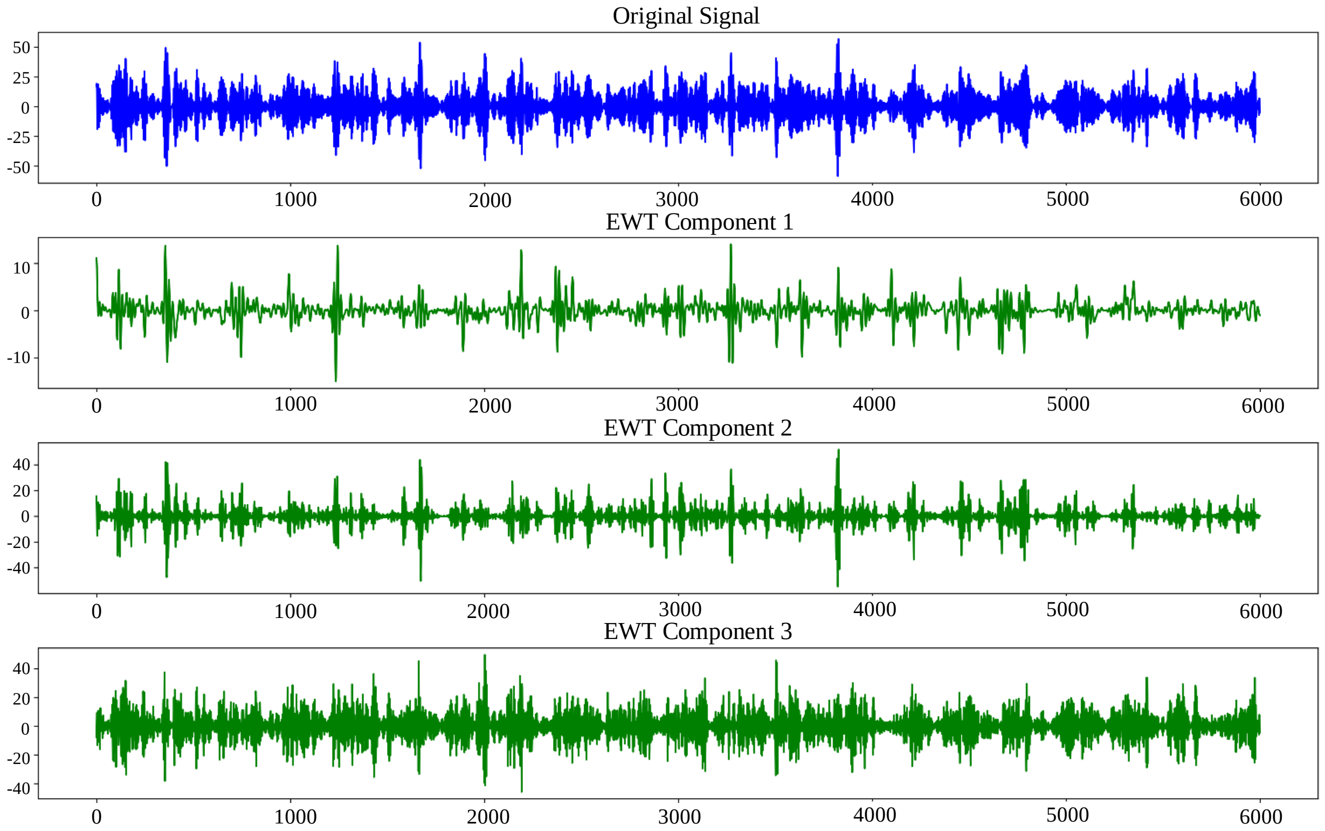


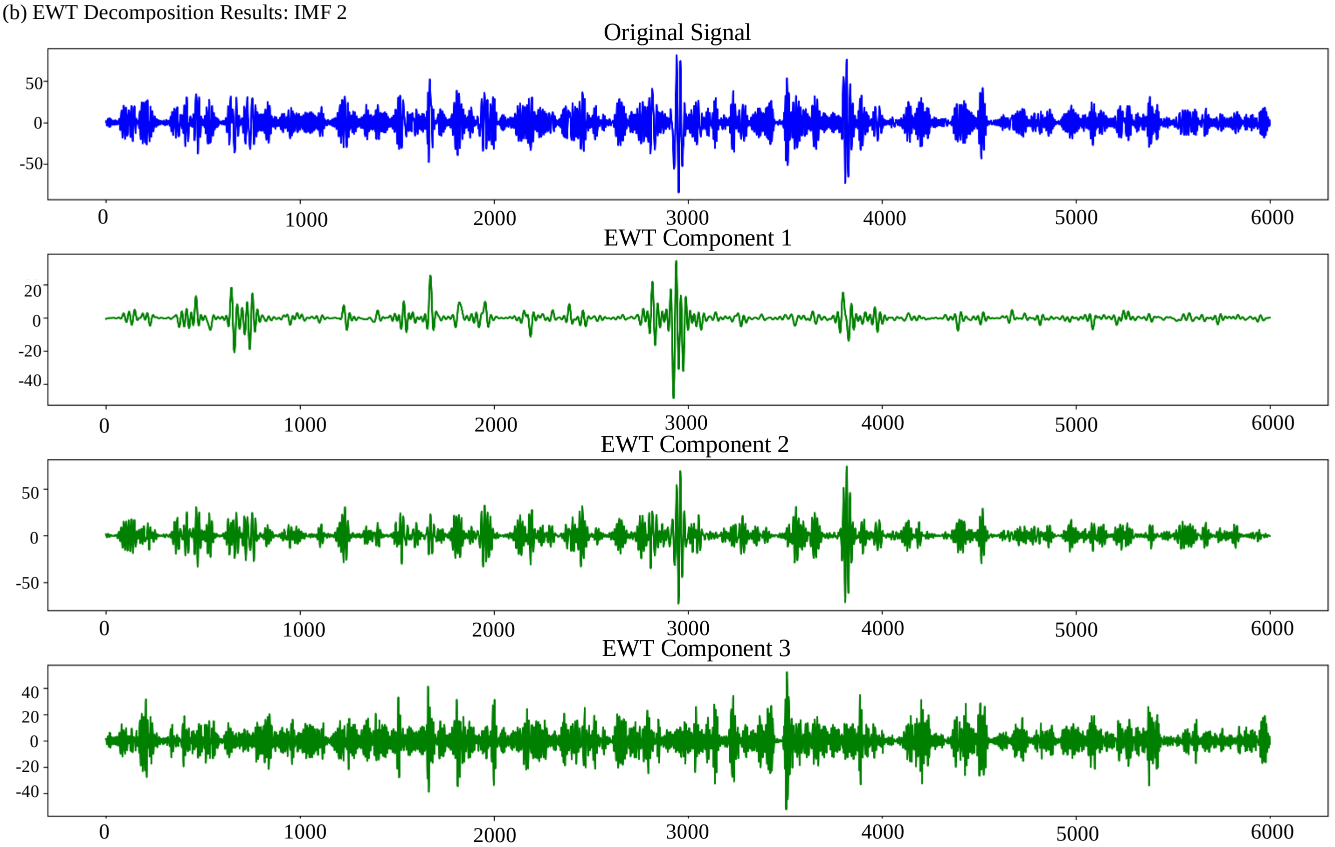
Fig 4. CEEMDAN Decomposition Results Diagram



(a) EWT Decomposition Results: IMF 1



(b) EWT Decomposition Results: IMF 2



(c) EWT Decomposition Results: IMF 3
Fig 5. EWT Decomposition

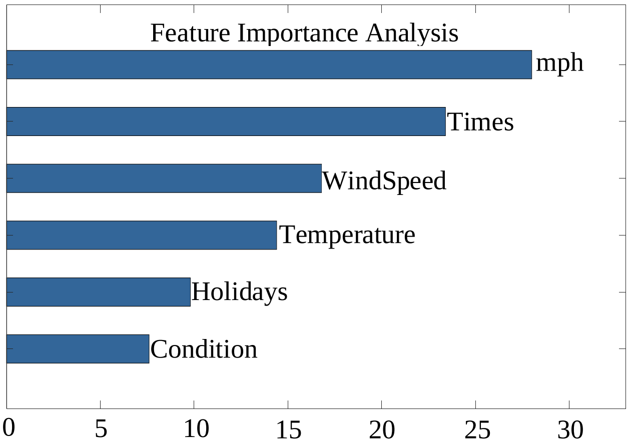


Fig 6. Feature Importance Analysis Diagram

C. Model Training and Testing

The hybrid neural network model is utilized for both training and prediction tasks. In the training phase, the SSA is employed to fine-tune the hyperparameters of the BiGRU-Attention model. The initial population size is configured to 50, with the iteration count set to 100.After 150 training epochs, SSA continuously updates the population positions during iterations, evaluates the quality of each individual using a fitness function, and dynamically adjusts the search strategy based on the global optimal solution. Ultimately, through multiple iterations and dynamic updates of population positions, SSA gradually approaches the global optimal solution. The findings indicate that the BiGRU-Attention model performs best with two layers and 44 hidden units per layer. By integrating a bidirectional GRU architecture with an attention mechanism, the model effectively captures temporal dependencies in the data sequence and dynamically emphasizes critical information points.

Fig 7 displays the MSE loss for both the training and test sets as the number of training epochs increases. The graph indicates that the loss for both sets decrease steadily, with the test set loss showing no significant rise, demonstrating the model's strong performance on both the training and test datasets.

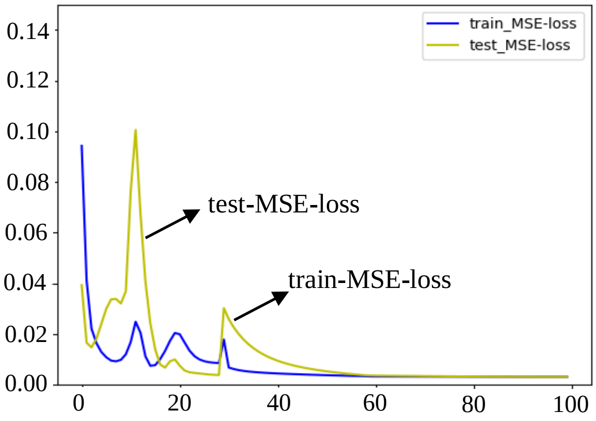


Fig 7. Loss Variation

D. Comparative Analysis of Results

The proposed multimodal decomposition-based hybrid neural network demonstrates consistent predictive superiority across two datasets with distinct spatiotemporal characteristics. As illustrated in Fig 8 and Fig 9, the model's predictions exhibit strong alignment with ground-truth observations under diverse traffic scenarios, including steady-state conditions, recurring peak-hour patterns, and irregular fluctuation events, validating its capability to holistically model the dynamic evolution mechanisms of traffic flow. Compared to standalone BiGRU, XGBoost, and conventional hybrid models, this framework achieves enhanced noise robustness and pattern generalization in heterogeneous data scenarios. Even when considering weather conditions and traffic scenarios, the multi-modal decomposition-based hybrid neural network model maintains a strong advantage in fitting abrupt changes in traffic flow data and minimizing prediction errors.

The experiments demonstrate that the hybrid neural network model, enhanced by data preprocessing and the inclusion of various external factors, substantially surpasses the individual BiGRU and XGBoost models in predictive accuracy. The correlation distributions between actual and predicted values for four models (BiGRU, XGBoost, BiGRU-XGBoost, and the multi-modal decomposition-based hybrid neural network) are compared in Fig 10, demonstrating the superior consistency of the hybrid model in capturing complex traffic patterns. By evaluating model performance using the R^2 correlation coefficient, it is evident that the traffic flow sampling data exhibits a continuous distribution from low to high values. The experimental outcomes reveal that the R^2 value of the hybrid neural network model, incorporating multi-modal decomposition, markedly surpasses that of the compared models, indicating superior predictive accuracy. Notably, compared to the standalone BiGRU and XGBoost models, the BiGRU-XGBoost model already shows a clear improvement in prediction accuracy, further confirming the advantages of hybrid models in traffic flow prediction.

TABLE IV and TABLE V summarize the overall performance metrics of different models on the training set and test set, respectively. As shown in TABLE IV, the hybrid model (CEEMDAN-EWT-SSA-BiGRU-XGBoost) achieves the lowest values in MAE, MSE, MAPE, and RMSE while maintaining the highest R^2 value, confirming its superior generalization capability. Notably, although inherent discrepancies exist in evaluation metrics between the two datasets, the hybrid model consistently outperforms baseline models across both tables, with observed variations attributable to the intrinsic spatiotemporal heterogeneity of traffic flow data and uncertainties caused by external disturbances. Furthermore, the proposed method demonstrates enhanced prediction accuracy and consistency compared to alternative models in both prediction outcomes and evaluation metrics, solidifying its practical value for traffic flow forecasting applications. Among all compared models on Dataset 1, the BiGRU-XGBoost hybrid model attained R^2, MSE, MAE, RMSE, and MAPE values of 0.8774, 0.403, 0.045, 0.501, and 0.176, respectively. Through enhanced data processing and integration of multiple external factors, the proposed model demonstrated superior performance with corresponding values of 0.9455, 0.293, 0.043, 0.334, and 0.126, validating its enhanced predictive capability in complex traffic scenarios.

The predictive performance of different models over extended forecasting horizons is compared in Fig 11, where the proposed hybrid model maintains higher R² values and lower MSE, MAE, RMSE, and MAPE across all time steps, demonstrating robust long-term forecasting capability.

To further elucidate how each module of our proposed method enhances prediction accuracy, we conducted corresponding ablation studies by incrementally incorporating submodules and comparing the resulting models. Starting with a baseline BiGRU-XGBoost model, we sequentially integrated the decomposition, multimodality, attention, and optimization modules, evaluating the final prediction performance. Fig 12 illustrates the predictions of these models, while TABLE VI provides a detailed comparison of performance metrics. As shown in TABLE VI, each submodule employed in this study significantly improves prediction accuracy. For instance, the baseline BiGRU-XGBoost model yields MAE, MSE, MAPE, RMSE, and R² values of 0.045, 0.403, 0.176, 0.501, and 0.8774, respectively. With the progressive incorporation of decomposition, multimodality, attention, and optimization modules, all metrics exhibit an optimization trend: MAE decreases to 0.043, MSE to 0.293, MAPE to 0.126, RMSE to 0.334, and R² increases to 0.9455, demonstrating the cumulative effectiveness of the proposed modules.

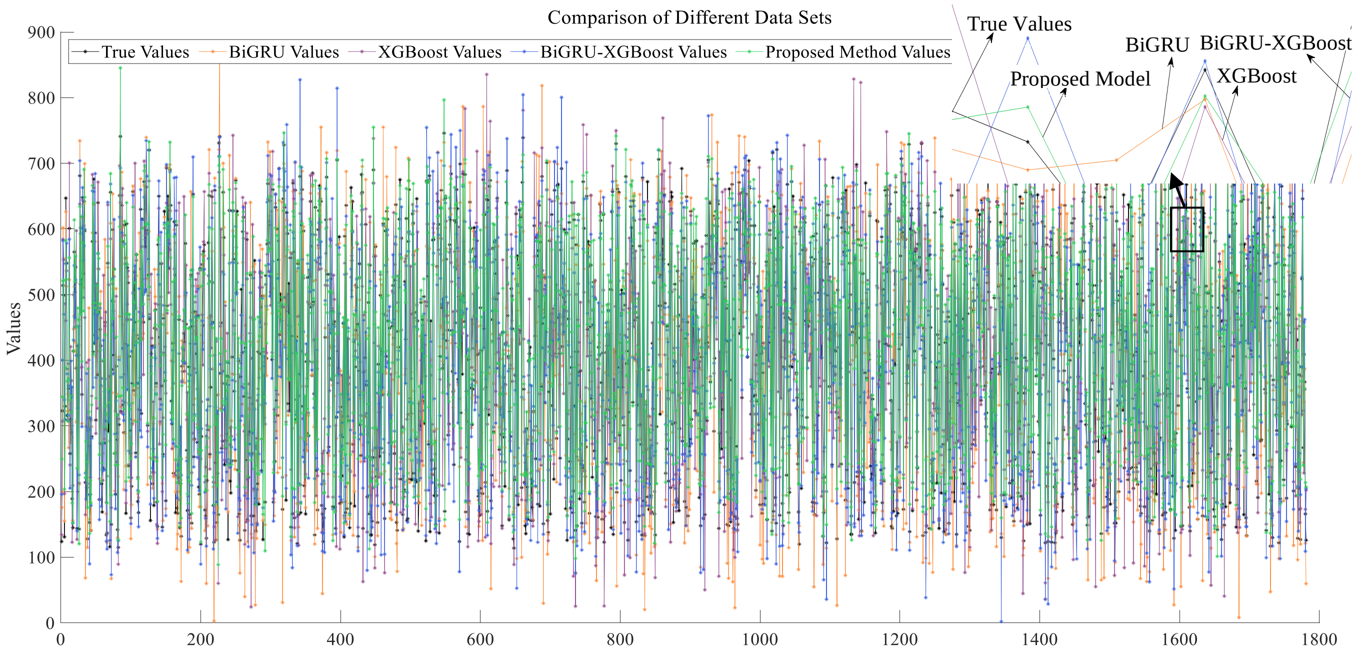


Fig 8. Comparison of Prediction Results for Different Models on Dataset 1

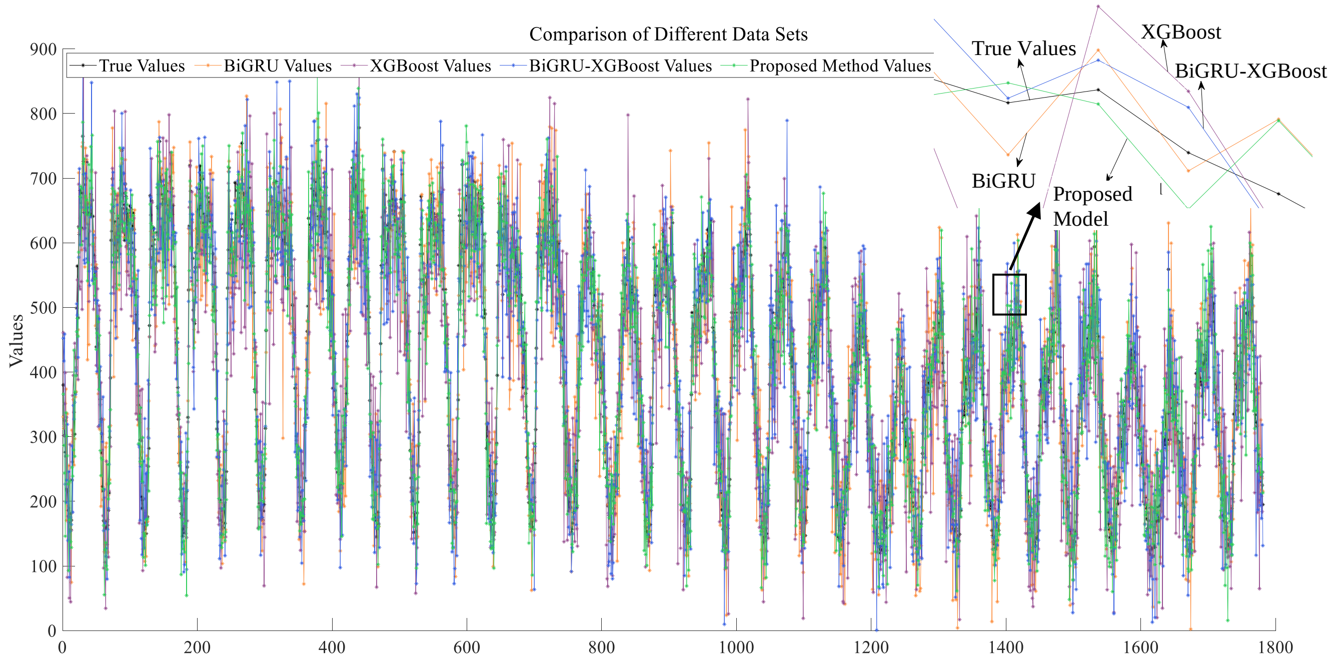


Fig 9. Comparison of Prediction Results for Different Models on Dataset 2

TABLE IV
COMPARISON OF EVALUATION METRICS FOR DIFFERENT MODELS ON DATASET 1

Methods	Training set				
	R²	MSE	MAE	RMSE	MAPE
BiGRU	0.8859	0.359	0.044	0.469	0.068
XGBoost	0.7777	0.700	0.061	0.648	0.089
BiGRU -XGBoost	0.8857	0.391	0.045	0.444	0.057
Proposed Model	0.9598	0.312	0.043	0.266	0.05
Methods	Test set				
	R²	MSE	MAE	RMSE	MAPE
BiGRU	0.8697	0.406	0.044	0.564	0.182
XGBoost	0.7574	0.755	0.064	0.677	0.192
BiGRU -XGBoost	0.8774	0.403	0.045	0.501	0.176
Proposed Model	**0.9455**	**0.293**	**0.043**	**0.334**	**0.126**

TABLE V
COMPARISON OF EVALUATION METRICS FOR DIFFERENT MODELS ON DATASET 2

Methods	Training set				
	R²	MSE	MAE	RMSE	MAPE
BiGRU	0.8602	0.400	0.145	0.703	0.068
XGBoost	0.7932	0.652	0.164	0.825	0.085
BiGRU -XGBoost	0.8653	0.391	0.148	0.691	0.063
Proposed Model	0.9222	0.310	0.144	0.546	0.052
Methods	Test set				
	R²	MSE	MAE	RMSE	MAPE
BiGRU	0.8522	0.423	0.148	0.778	0.187
XGBoost	0.7774	0.735	0.166	0.868	0.191
BiGRU -XGBoost	0.8598	0.410	0.146	0.610	0.180
Proposed Model	**0.9147**	**0.320**	**0.145**	**0.556**	**0.135**

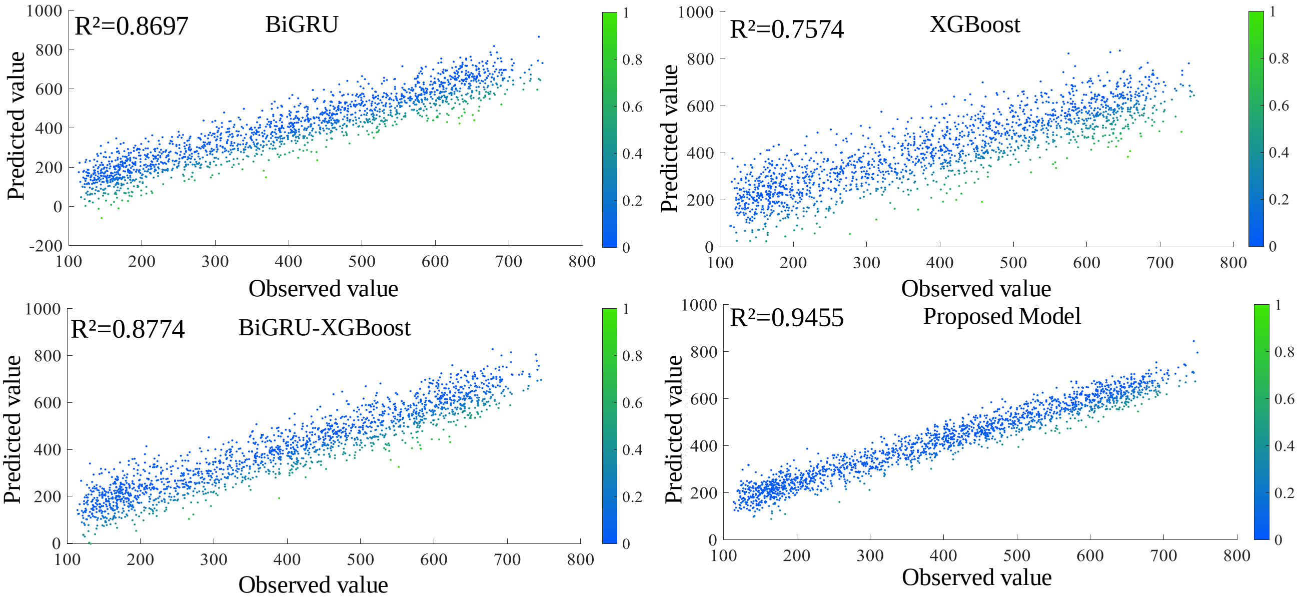


Fig 10. Comparative Analysis of Prediction Results from Different Models

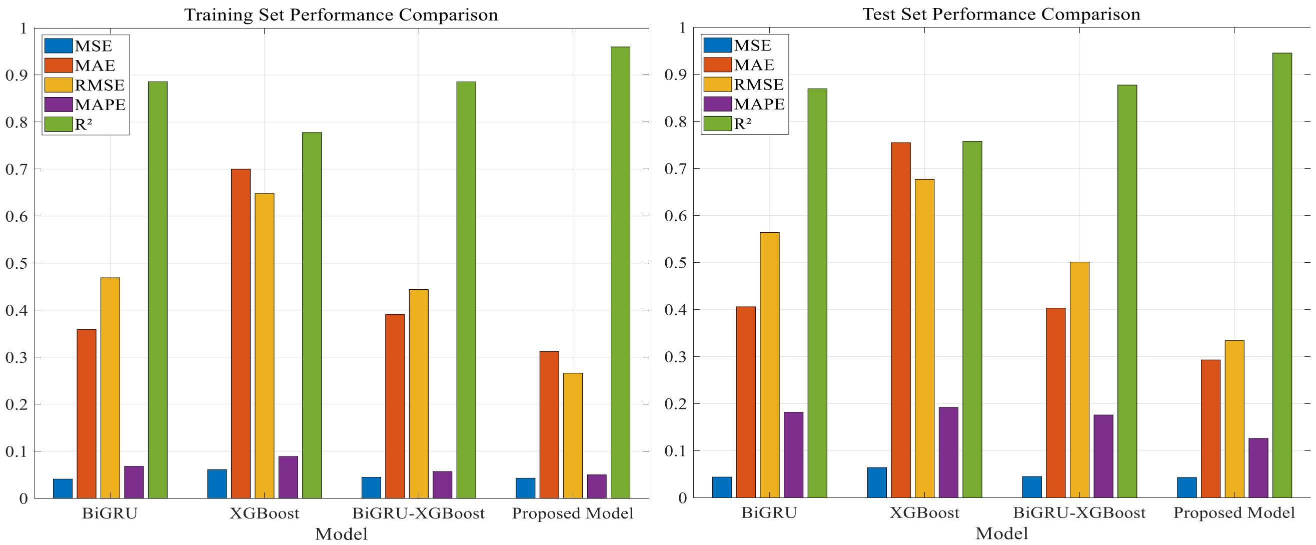


Fig 11. Predictive Performance of Different Models over Extended Forecasting Horizons

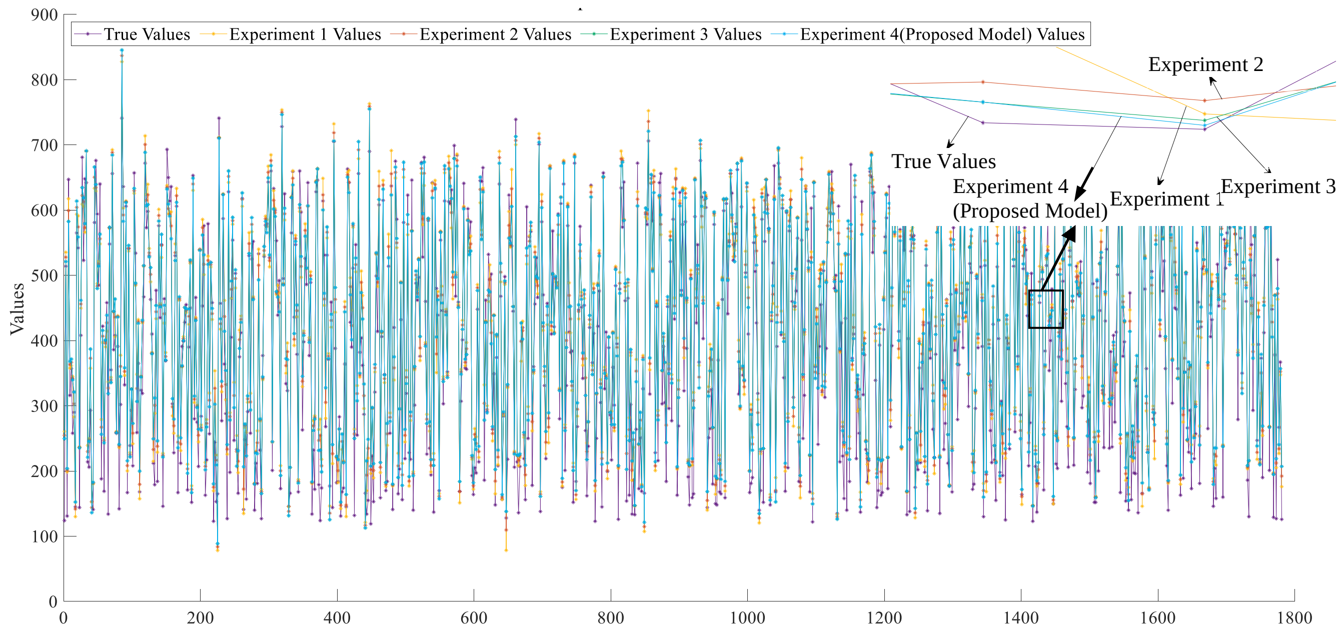


Fig 12. Performance Comparison of Baseline Models and Ablation Studies

TABLE VI
PERFORMANCE COMPARISON OF BASELINE MODELS AND ABLATION STUDIES

Methods	DECOMPOSITION	MULTIMODALITY	ATTENTION	OPTIMIZATION	R²	MSE	MAE	RMSE	MAPE
BiGRU -XGBoost					0.8774	0.403	0.045	0.501	0.176
Experiment 1	√				0.9081	0.351	0.044	0.456	0.161
Experiment 2	√	√			0.9202	0.323	0.044	0.412	0.146
Experiment 3	√	√	√		0.9354	0.300	0.043	0.365	0.133
Experiment 4 (Proposed Model)	√	√	√	√	0.9455	0.293	0.043	0.334	0.126

Experimental results validate the effectiveness of the hybrid neural network model in traffic flow prediction. The model demonstrates not only higher R² values and lower MSE, MAE, RMSE, and MAPE on the test set but also maintains superior performance on the training set. Comparative analysis with benchmark models further confirms the reliability and robustness of the CEEMDAN-EWT-based hybrid neural model in processing dynamic traffic flow data.

V. Conclusion

This study significantly improves the accuracy and robustness of traffic flow prediction by proposing a hybrid neural network model based on CEEMDAN-EWT. The CEEMDAN algorithm plays a critical role in the data preprocessing stage, effectively removing noise and extracting the main components of traffic flow data through adaptive decomposition. The EWT further decomposes high-frequency IMFs to reduce noise and extract more meaningful features, thereby enhancing the model's ability to capture dynamic changes in traffic flow.

The SSA optimizes model parameters by simulating the social behavior of sparrows, intelligently searching the solution space to determine the optimal number of hidden layers and neurons per layer, further improving model performance. During the feature extraction stage, the BiGRU-Attention model effectively captures dynamic dependencies and key information points in time series by combining a bidirectional GRU structure and an attention mechanism, enhancing prediction accuracy. In the refinement prediction stage, the XGBoost model adjusts the output of the BiGRU-Attention model using its excellent generalization ability and capability to handle nonlinear problems, further improving prediction accuracy. This step refines the prediction results, significantly enhancing the model's predictive performance.

In comparative evaluations, the proposed CEEMDAN-based hybrid neural network model demonstrates higher prediction accuracy and robustness on the test set compared to standalone BiGRU and XGBoost models. Specifically, the R² value improved by approximately **8.72%**, **24.84%**, and **7.76%** higher than that of the BiGRU, XGBoost, and BiGRU-XGBoost models, respectively. The MSE value is reduced by approximately **27.83%**, **61.20%**, and **27.29%**, respectively, the MAE value is reduced by approximately **2.27%**, **32.81%**, and **4.44%**, the RMSE value is reduced by approximately **40.78%**, **50.66%**, and **33.33%**, and the MAPE value is reduced by approximately **30.77%**, **34.38%**, and **28.42%**respectively. These results highlight the advantages of the proposed model in prediction accuracy and error reduction. In addition, to rigorously evaluate the contribution of each submodule to model accuracy, we performed systematic ablation studies. The comparison outcomes demonstrate that all evaluation metrics consistently improve, confirming the effectiveness and robustness of the proposed enhancements.

These findings clearly demonstrate the significant advantages of the proposed model in prediction precision. By integrating multiple techniques and methods, the model not only achieves an R² value close to perfection but also significantly reduces key performance metrics such as MSE, MAE, RMSE, and MAPE. This indicates that the model can effectively capture the dynamic characteristics of traffic flow, improving the reliability and accuracy of predictions.

REFERENCES

[1] Vlahogianni, Eleni I., Matthew G. Karlaftis, and John C. Golias. "Short-term traffic forecasting: Where we are and where we’re going." *Transportation Research Part C: Emerging Technologies* 43 (2014): 3-19.

[2] Kumar, S. Vasantha, and Lelitha Vanajakshi. "Short-term traffic flow prediction using seasonal ARIMA model with limited input data." *European Transport Research Review* 7 (2015): 1-9.

[3] Yang, Hao-Fan, et al. "Optimized configuration of exponential smoothing and extreme learning machine for traffic flow forecasting." *IEEE Transactions on Industrial Informatics* 15.1 (2018): 23-34.

[4] Kumar, Selvaraj Vasantha. "Traffic flow prediction using Kalman filtering technique." *Procedia Engineering* 187 (2017): 582-587.

[5] Pun, Lilian, Pengxiang Zhao, and Xintao Liu. "A multiple regression approach for traffic flow estimation." *IEEE Access* 7 (2019): 35998-36009.

[6] Gurcan, C., B. Negash, and H. Nathan. "Improved grey system models for predicting traffic parameters." *Expert Syst. Appl* 177 (2021): 114972.

[7] Hamed, Mohammad M., Hashem R. Al-Masaeid, and Zahi M. Bani Said. "Short-term prediction of traffic volume in urban arterials." *Journal of Transportation Engineering* 121.3 (1995): 249-254.

[8] Kumar, Praveen B., and K. Hariharan. "Time series traffic flow prediction with hyper-parameter optimized ARIMA models for intelligent transportation system." *Journal of Scientific & Industrial Research* 81.04 (2022): 408-415.

[9] Yang, Hanyu, et al. "A network traffic forecasting method based on SA optimized ARIMA–BP neural network." *Computer Networks* 193 (2021): 108102.

[10] Williams, Billy M., and Lester A. Hoel. "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results." *Journal of Transportation Engineering* 129.6 (2003): 664-672.

[11] Yang, Hanyu, et al. "A network traffic forecasting method based on SA optimized ARIMA–BP neural network." *Computer Networks* 193 (2021): 108102.

[12] Shoaeinaeini, Maryam, Oktay Ozturk, and Deepak Gupta. "Twitter-informed Prediction for Urban Traffic Flow Using Machine Learning." *2022 6th International Conference on Universal Village (UV)*. IEEE, 2022.

[13] Chi, Zengxiao, and Lin Shi. "Short-term traffic flow forecasting using ARIMA-SVM algorithm and R." *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*. IEEE, 2018.

[14] Hong, Wei-Chiang, et al. "Hybrid evolutionary algorithms in a SVR traffic flow forecasting model." *Applied Mathematics and Computation* 217.15 (2011): 6733-6747.

[15] Wang, Jin, and Qixin Shi. "Short-term traffic speed forecasting hybrid model based on chaos–wavelet analysis-support vector machine theory." *Transportation Research Part C: Emerging Technologies* 27 (2013): 219-232.

[16] Zhan, Aiyun, et al. "A traffic flow forecasting method based on the GA-SVR." *Journal of High Speed Networks* 28.2 (2022): 97-106.

[17] Nikravesh, Ali Yadavar, et al. "Mobile network traffic prediction using MLP, MLPWD, and SVM." *2016 IEEE International Congress on Big Data (BigData Congress)*. IEEE, 2016.

[18] Lv, Yisheng, et al. "Traffic flow prediction with big data: A deep learning approach." *IEEE Transactions on Intelligent Transportation Systems* 16.2 (2014): 865-873.

[19] Vinayakumar, Ravi, K. P. Soman, and Prabaharan Poornachandran. "Applying deep learning approaches for network traffic prediction." *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2017.

[20] Yin, Xueyan, et al. "Deep learning on traffic prediction: Methods, analysis, and future directions." *IEEE Transactions on Intelligent Transportation Systems* 23.6 (2021): 4927-4943.

[21] Tedjopurnomo, David Alexander, et al. "A survey on modern deep neural network for traffic prediction: Trends, methods and challenges." *IEEE Transactions on Knowledge and Data Engineering* 34.4 (2020): 1544-1561.

[22] Zhao, Zheng, et al. "LSTM network: a deep learning approach for short-term traffic forecast." *IET Intelligent Transport Systems* 11.2 (2017): 68-75.

[23] Li, Zonghan, et al. "Adaptive data processing framework for efficient short-term traffic flow prediction." *Nonlinear Dynamics* 112.17 (2024): 15231-15249.

[24] Wu, Yuankai, et al. "A hybrid deep learning based traffic flow prediction method and its understanding." *Transportation Research Part C: Emerging Technologies* 90 (2018): 166-180.

[25] Du, Shengdong, et al. "A hybrid method for traffic flow forecasting using multimodal deep learning." *International Journal of Computational Intelligence Systems* 13.1 (2020): 85-97.

[26] Wang, Ke, et al. "A hybrid deep learning model with 1DCNN-LSTM-Attention networks for short-term traffic flow prediction." *Physica A: Statistical Mechanics and its Applications* 583 (2021): 126293.

[27] Zheng, Haifeng, et al. "A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction." *IEEE Transactions on Intelligent Transportation Systems* 22.11 (2020): 6910-6920.

[28] Harrou, Fouzi, et al. "Enhancing road traffic flow prediction with improved deep learning using wavelet transforms." *Results in Engineering* 23 (2024): 102342.

[29] Mousavizadeh Kashi, Seyed Omid, and Meisam Akbarzadeh. "A framework for short-term traffic flow forecasting using the combination of wavelet transformation and artificial neural networks." *Journal of Intelligent Transportation Systems* 23.1 (2019): 60-71.

[30] Lu, Wenqi, et al. "A hybrid model for lane-level traffic flow forecasting based on complete ensemble empirical mode decomposition and extreme gradient boosting." *IEEE Access* 8 (2020): 42042-42054.

[31] Wu, Xinye, Shude Fu, and Zujie He. "Research on short-term traffic flow combination prediction based on CEEMDAN and machine learning." *Applied Sciences* 13.1 (2022): 308.