fractal and fractional

MDPI

Article

A Fractional-Prior Anchored Extreme Learning Machine for Short-Term Traffic Flow Forecasting

Junhao Ye [1], Zhuoxuan Li [2,3,*], Shaosheng Xu [4] and Jinde Cao [5,*]

[1] College of Information Science and Engineering, Hohai University, Changzhou 213200, China
[2] College of Civil and Transportation Engineering, Hohai University, Nanjing 210098, China
[3] Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland
[4] School of Computer and Information, Anhui Normal University, Wuhu 241000, China
[5] School of Mathematics, Southeast University, Nanjing 210096, China
* Correspondence: zhuoxuan.li@hhu.edu.cn (Z.L.); jdcao@seu.edu.cn (J.C.)

Abstract

Short-term traffic flow forecasting must account for persistent and recurrent patterns while representing local fluctuations. This study proposes a Fractional-Prior Anchored Extreme Learning Machine (FPA-ELM) for station-level forecasting. A persistence–seasonal anchor combines the latest observation with historical flows at the same seasonal position as the forecast target. Normalized Grünwald–Letnikov-type accumulation coefficients provide a soft prior for the seasonal coefficient profile, while the persistence coefficient and overall seasonal contribution are estimated from data. An ELM with fixed random features and direct linear connections learns the remaining residuals. Both stages are fitted by solving regularized linear systems without back-propagation. Evaluated against 21 baselines on 36 traffic-flow series sampled every 15 min, FPA-ELM achieves the lowest station-averaged mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE) for one-step forecasting, at 44.48, 66.42, and 8.92%, respectively. Relative to XGBoost, the strongest baseline in MAE and RMSE, the corresponding reductions are 5.49% and 4.59%. Ablation results support the complementary roles of the anchor and residual learner. Window sensitivity further indicates that the preferred lag length depends on the forecasting horizon. These findings support separating structured temporal information from nonlinear residual correction for lightweight traffic flow forecasting.

Keywords: traffic flow forecasting; fractional prior; fractional accumulation; extreme learning machine; residual learning

Check for updates

Academic Editor: Haci Mehmet Baskonus

Received: 15 October 2026
Revised: 12 November 2026
Accepted: 16 November 2026
Published: 20 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Short-term traffic flow forecasts are used to support traffic management, congestion mitigation, and transport operations in Intelligent Transportation Systems (ITS). Roadside sensors and connected infrastructure provide increasingly extensive traffic measurements, supporting the development of data-driven forecasting methods. For individual monitoring stations, predictive accuracy must be balanced against the computational resources available. Lightweight models are therefore needed to capture temporal structure without complex network architectures or extensive iterative training.

Statistical approaches, including autoregressive models, exponential smoothing, and Kalman filtering, offer computationally efficient descriptions of temporal evolution [1–3]. Their assumptions, however, can restrict the representation of nonlinear fluctuations and

changing traffic conditions. More flexible representations have been developed using recurrent networks, convolutional architectures, and attention-based models [4–6]. Fractional-order recurrent networks and time-varying fractional autoregressive models have also been developed to capture temporal dependencies and nonstationary behaviour [7,8]. Many such methods require iterative optimization and substantial computational resources. For station-level forecasting from limited historical sequences, greater complexity does not necessarily improve predictive performance. This raises a practical question: how much temporal structure can be represented explicitly before nonlinear learning?

Traffic flow combines short-term persistence with recurring variation across seasonal cycles. At short horizons, the latest observation often provides an informative estimate of the future traffic level. Observations at corresponding positions in earlier cycles provide additional information about recurrent demand. Their relevance can differ because traffic conditions change between cycles. Seasonal information is omitted from persistence forecasting, whereas historical cycles are weighted equally in a conventional seasonal average. Estimating each seasonal coefficient independently allows greater flexibility but can be sensitive to noise and correlations among predictors. Historical dependence and seasonality have also been examined through seasonal fractional integration and fractional-order seasonal forecasting in other domains [9,10]. These studies motivate temporal representations that distinguish the influence of earlier cycles while adapting the seasonal contribution to observed data.

Fractional calculus represents history-dependent processes through nonlocal operators whose order controls the relative influence of historical states [11]. In discrete time, a structured weighting profile for fractional accumulation is defined by generalized binomial coefficients. Fractional-order accumulative generation was interpreted through discrete convolution by Chen [12], linking fractional accumulation to sequence modelling. Accumulation has also been incorporated into fractional-order grey prediction models to represent historical information with adjustable weights [13,14]. For traffic forecasting, these coefficients offer a way to represent seasonal history beyond the latest observation or a uniform average.

The question is how to use fractional weighting without imposing an overly restrictive seasonal representation. Fractional-order grey models commonly transform the original sequence through accumulation before estimation. Extensions introduce dynamic fractional orders, neural networks, or metaheuristic optimization to increase flexibility [14,15]. These procedures differ from using fractional coefficients as prior information in regression. A fixed fractional profile may be unsuitable when the relevance of seasonal observations varies across stations and operating conditions. Independent coefficient estimation, however, discards the structure supplied by that profile. A soft regularization prior allows seasonal coefficients to depart from the fractional weights when supported by the training data. The fractional order specifies the prior distribution of historical influence, while the data determine the overall seasonal contribution. This formulation uses fractional structure without assuming that traffic flow satisfies a fractional differential equation.

An extreme learning machine (ELM) provides a computationally efficient learner for the nonlinear component of traffic flow. Its randomly generated hidden-layer parameters are fixed, and output weights are estimated analytically without repeated backpropagation [16,17]. Its representation capability has been extended through modifications to architecture, initialization, and regularization [18–20]. Direct ELM forecasting nevertheless requires the model to learn both the dominant traffic level and deviations around it. Because the hidden representation is fixed, learning regularities already captured by persistence and seasonal observations may consume capacity needed for those deviations. Fractional-order dynamics have also been combined with machine-learning residual correction in hybrid forecasting studies, suggesting complementarity between structured

temporal models and nonlinear error learning [21]. Their fractional formulations and nonlinear learners differ from those considered here. For lightweight station-level forecasting, a persistence–seasonal anchor provides a simple reference around which an ELM can learn residual variation.

This study aims to develop a lightweight and accurate framework for station-level traffic flow forecasting by integrating persistent and seasonal patterns with nonlinear residual learning. The objectives are to assess the effectiveness of fractional-prior regularization against conventional seasonal weighting schemes, examine the contributions of the anchor and residual learner, and evaluate forecasting accuracy and computational efficiency across different prediction horizons and input-window configurations.

To achieve these objectives, we propose a Fractional-Prior Anchored Extreme Learning Machine (FPA-ELM). The framework first constructs a persistence–seasonal anchor using the latest observation and historical flows at the corresponding seasonal positions. The seasonal coefficients are regularized towards a normalized Grünwald–Letnikov-type fractional accumulation profile, while their overall contribution and the persistence coefficient are estimated from data. An ELM then learns the remaining residuals from recent observations, calendar information, and local traffic statistics. Its fixed random hidden features and direct linear connections are combined through a regularized readout. Both stages are estimated using regularized linear systems without iterative back-propagation.

The framework separates structured temporal information from residual fluctuations, allowing the ELM to focus on variations not captured by the anchor. Its forecasting performance is assessed on 36 traffic-flow series through baseline comparisons, ablation experiments, sensitivity analyses, and computational efficiency evaluation. The main contributions are summarized as follows:

- A fractional seasonal-memory prior is developed using normalized Grünwald–Letnikov-type accumulation coefficients, providing a structured and adjustable weighting profile for historical seasonal observations.
- A persistence–seasonal anchor is formulated through regularized least squares, allowing the seasonal coefficients to adapt to observed data while retaining the fractional weighting structure.
- A lightweight forecasting model is developed by integrating the anchor with ELM residual learning. Fixed random features, direct linear connections, and separate lag and context windows support efficient estimation and flexible temporal representation.
- Extensive experiments on 36 real-world traffic-flow series sampled at 15-min intervals evaluate forecasting accuracy, computational efficiency, and the contributions of individual model components across different forecasting horizons.

Relevant studies on traffic forecasting, fractional-order time-series modelling, and lightweight neural learning are reviewed in the next section. The fractional-prior anchor, ELM residual correction, and estimation procedure are then presented. Subsequent sections report the experimental settings, comparative results, and mechanism analyses. The method's applicability and limitations are examined in the discussion, followed by the conclusions.

2. Fractional-Prior Anchored Extreme Learning Machine for Traffic Flow Forecasting

The fractional-prior anchored extreme learning machine (FPA-ELM) combines a persistence–seasonal anchor with an extreme learning machine (ELM) residual correction, as shown in Figure 1. The fractional prior regularizes the relative profile of the seasonal coefficients, while the ELM models the remaining deviations using fixed random hidden features and a regularized readout.

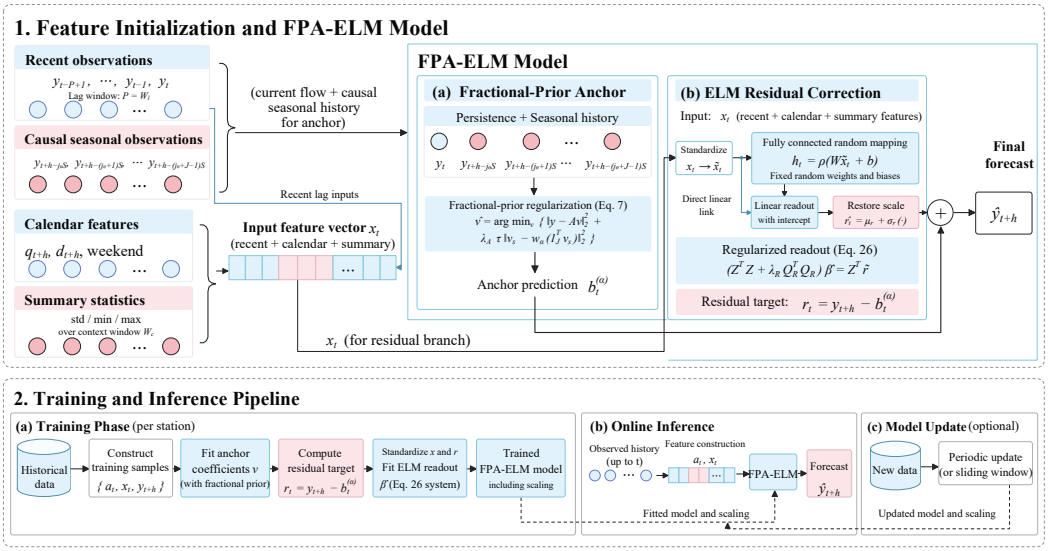


Figure 1. Architecture of FPA-ELM for traffic flow forecasting. The upper panel shows (a) the fractional-prior anchor and (b) the ELM residual correction. The lower panel presents (a) the station-wise training phase, (b) online inference, and (c) optional model updating. Light-blue and pink colors visually distinguish feature groups and processing elements without encoding numerical values.

2.1. Forecasting Problem Formulation

Let $y_{i,t}$ denote the traffic flow observed at station i during sampling interval t. At forecast origin t, the task is to estimate $y_{i,t+h}$ for a prescribed horizon $h \geq 1$. Models are fitted separately for each station, and the station index is omitted below unless required.

The predictors comprise current and recent flows, observations from earlier seasonal cycles, and calendar information associated with the forecast target. Flow predictors are drawn from observations available at or before t, while target-time calendar variables are known at the forecast origin. The seasonal period S and forecast horizon h are measured in sampling intervals. For each forecast origin, FPA-ELM produces a scalar prediction $\widehat{y}_{t+h}$ as the sum of an anchor component and a residual correction.

2.2. Fractional-Prior Anchor

The anchor combines the current flow with historical observations at the same seasonal position as the forecast target. The seasonal history contains only observations available at the forecast origin. For a seasonal period of S sampling intervals, the first admissible seasonal lag is

$$j_0 = \max\left(1, \left\lceil \frac{h}{S} \right\rceil\right). \tag{1}$$

Starting from this lag, the history vector collects J seasonal observations:

$$\mathbf{s}_t = \left(y_{t+h-j_0 S}, y_{t+h-(j_0+1)S}, \ldots, y_{t+h-(j_0+J-1)S}\right)^\top \in \mathbb{R}^J. \tag{2}$$

The offset j_0 ensures that all seasonal observations occur at or before t. Combining the seasonal history with the persistence observation gives the anchor input

$$\mathbf{a}_t = \begin{bmatrix} p_t \\ \mathbf{s}_t \end{bmatrix}, \qquad p_t = y_t. \tag{3}$$

For N training origins $t_1, \ldots, t_N$, the anchor inputs form the design matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_{t_1}^\top \\ \vdots \\ \mathbf{a}_{t_N}^\top \end{bmatrix} = \begin{bmatrix} \mathbf{p} & \mathbf{S} \end{bmatrix} \in \mathbb{R}^{N\times(J+1)}, \tag{4}$$

where $\mathbf{p}$ contains the persistence observations and $\mathbf{S}$ contains the seasonal predictors. The target vector is

$$\mathbf{y} = (y_{t_1+h}, \ldots, y_{t_N+h})^\top.$$

The corresponding coefficients are

$$\mathbf{v} = (v_p, \mathbf{v}_s^\top)^\top \in \mathbb{R}^{J+1},$$

where v_p weights the persistence term and $\mathbf{v}_s \in \mathbb{R}^J$ weights the seasonal history.

The relative seasonal coefficient profile is regularized using a normalized Grünwald–Letnikov-type kernel. For fractional order α, the kernel coefficients satisfy

$$\omega_0 = 1, \qquad \omega_k = \omega_{k-1}\frac{\alpha+k-1}{k}, \qquad k = 1, \ldots, J-1. \tag{5}$$

Defining $\kappa_\alpha(k) = \omega_k$, the normalized fractional weights are

$$w_{\alpha,j} = \frac{\kappa_\alpha(j-1)}{\sum_{\ell=0}^{J-1}\kappa_\alpha(\ell)}, \qquad j = 1, \ldots, J. \tag{6}$$

The resulting vector

$$\mathbf{w}_\alpha = (w_{\alpha,1}, \ldots, w_{\alpha,J})^\top$$

satisfies

$$\mathbf{1}_J^\top \mathbf{w}_\alpha = 1,$$

with entries ordered from the most recent admissible seasonal observation to the oldest. At $\alpha = 1$, the weights are uniform; $\alpha < 1$ places greater weight on recent cycles, whereas $\alpha > 1$ shifts weight toward earlier cycles.

The fractional profile regularizes the relative seasonal coefficients while the overall seasonal contribution is estimated from the data. The anchor coefficients are obtained from

$$\widehat{\mathbf{v}} = \arg\min_{\mathbf{v}}\left\{ \|\mathbf{y} - \mathbf{A}\mathbf{v}\|_2^2 + \lambda_A \tau \left\| \mathbf{v}_s - \mathbf{w}_\alpha\left(\mathbf{1}_J^\top \mathbf{v}_s\right)\right\|_2^2 \right\}, \tag{7}$$

with the training-data scale

$$\tau = \frac{1}{J}\|\mathbf{S}\|_F^2. \tag{8}$$

The parameter λ_A controls the prior strength, and τ scales the penalty according to the magnitude of the seasonal predictors. The persistence coefficient and the total seasonal contribution $\mathbf{1}_J^\top \mathbf{v}_s$ are estimated directly from the data, while the fractional prior regularizes the relative distribution of the seasonal coefficients.

For estimation, define

$$\mathbf{B}_\alpha = \mathbf{I}_J - \mathbf{w}_\alpha \mathbf{1}_J^\top, \tag{9}$$

and

$$\mathbf{M}_\alpha = \mathbf{B}_\alpha^\top \mathbf{B}_\alpha.$$

The complete penalty matrix is

$$\mathbf{P}_\alpha = \begin{bmatrix} 0 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{M}_\alpha \end{bmatrix}. \tag{10}$$

The zero first row and column exclude the persistence coefficient from the fractional-prior penalty. A small diagonal jitter is included to stabilize the linear system under near-collinearity:

$$\left(\mathbf{A}^\top \mathbf{A} + \lambda_A \tau \mathbf{P}_\alpha + \varepsilon \mathbf{I}\right)\widehat{\mathbf{v}} = \mathbf{A}^\top \mathbf{y}, \qquad \varepsilon > 0. \tag{11}$$

Solving this system yields the fitted anchor prediction

$$b_t^{(\alpha)} = \mathbf{a}_t^\top \widehat{\mathbf{v}}. \tag{12}$$

2.3. ELM Residual Correction

After fitting the anchor, the ELM models the residual target

$$r_t = y_{t+h} - b_t^{(\alpha)}. \tag{13}$$

The residual predictors combine recent traffic flows, target-time calendar information, and local traffic-state summaries. The P most recent observations form the lag block

$$\mathbf{x}_t^{(\mathrm{lag})} = \left(y_{t-P+1}, \ldots, y_t\right)^\top \in \mathbb{R}^P. \tag{14}$$

For the input-window analysis, the lag length is denoted by $W_l = P$. Local statistics are calculated from an independent context block

$$\mathbf{x}_t^{(\mathrm{ctx})} = \left(y_{t-W_c+1}, \ldots, y_t\right)^\top \in \mathbb{R}^{W_c}, \tag{15}$$

so W_l and W_c control different temporal components of the residual input. The default configuration uses

$$W_l = W_c = 12.$$

The calendar variables q_{t+h} and d_{t+h} denote the target's within-period position and day-of-week index, respectively. Their periodic encodings are combined with the standard deviation, minimum, and maximum of the context block:

$$\begin{aligned} \boldsymbol{\phi}_{t+h} = \Big[& \sin\!\left(\frac{2\pi q_{t+h}}{S}\right), \cos\!\left(\frac{2\pi q_{t+h}}{S}\right), \\ & \sin\!\left(\frac{4\pi q_{t+h}}{S}\right), \cos\!\left(\frac{4\pi q_{t+h}}{S}\right), \\ & \sin\!\left(\frac{2\pi d_{t+h}}{7}\right), \cos\!\left(\frac{2\pi d_{t+h}}{7}\right), \mathbb{I}_{\mathrm{weekend}}(d_{t+h}), \\ & \mathrm{std}\!\left(\mathbf{x}_t^{(\mathrm{ctx})}\right), \min\!\left(\mathbf{x}_t^{(\mathrm{ctx})}\right), \max\!\left(\mathbf{x}_t^{(\mathrm{ctx})}\right) \Big]^\top. \end{aligned} \tag{16}$$

Appending these features to the lag block gives the complete residual input:

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{x}_t^{(\mathrm{lag})} \\ \boldsymbol{\phi}_{t+h} \end{bmatrix} \in \mathbb{R}^D, \qquad D = P + 10. \tag{17}$$

Before ELM fitting, the inputs and residual targets are standardized using statistics estimated from the corresponding training data. Let $\boldsymbol{\mu}_x$ and $\boldsymbol{\sigma}_x$ denote the feature-wise input means and standard deviations, and let μ_r and σ_r denote the residual statistics. The standardized variables are

$$\widetilde{\mathbf{x}}_t = (\mathbf{x}_t - \boldsymbol{\mu}_x) \oslash \boldsymbol{\sigma}_x, \qquad \widetilde{r}_t = \frac{r_t - \mu_r}{\sigma_r}, \tag{18}$$

where $\oslash$ denotes element-wise division. Zero or near-zero feature standard deviations are replaced by one before normalization using the implementation tolerance.

The standardized input is mapped to L hidden features through activation function ρ:

$$\mathbf{h}_t = \rho(\mathbf{W}\widetilde{\mathbf{x}}_t + \mathbf{b}), \qquad \mathbf{W} \in \mathbb{R}^{L\times D}, \quad \mathbf{b} \in \mathbb{R}^L. \tag{19}$$

The hidden weights and biases are generated from fixed random streams:

$$\mathbf{W}_{\ell,:} \sim \mathcal{N}\left(\mathbf{0}, \frac{1}{D}\mathbf{I}_D\right), \qquad b_\ell \sim \mathcal{U}(-1,1), \qquad \ell = 1,\ldots,L. \tag{20}$$

These parameters remain fixed during readout estimation. The readout combines the hidden features, standardized inputs, and an output intercept. The standardized inputs form a direct linear path for residual correction:

$$\mathbf{z}_t = \begin{bmatrix} \mathbf{h}_t \\ \widetilde{\mathbf{x}}_t \\ 1 \end{bmatrix} \in \mathbb{R}^K, \qquad K = L + D + 1. \tag{21}$$

At the training origins, the design vectors and standardized residual targets are collected as

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z}_{t_1}^\top \\ \vdots \\ \mathbf{z}_{t_N}^\top \end{bmatrix} \in \mathbb{R}^{N\times K}, \tag{22}$$

and

$$\widetilde{\mathbf{r}} = \begin{bmatrix} \widetilde{r}_{t_1} \\ \vdots \\ \widetilde{r}_{t_N} \end{bmatrix}. \tag{23}$$

The readout coefficients are estimated by

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{ \|\mathbf{Z}\boldsymbol{\beta} - \widetilde{\mathbf{r}}\|_2^2 + \lambda_R \|\mathbf{Q}_R\boldsymbol{\beta}\|_2^2 \right\}. \tag{24}$$

The penalty matrix regularizes the hidden-feature and direct-link coefficients while leaving the output intercept unpenalized:

$$\mathbf{Q}_R = \mathrm{diag}(1,\ldots,1,0). \tag{25}$$

For readout regularization λ_R, the coefficient vector satisfies

$$\left(\mathbf{Z}^\top\mathbf{Z} + \lambda_R\mathbf{Q}_R^\top\mathbf{Q}_R\right)\widehat{\boldsymbol{\beta}} = \mathbf{Z}^\top\widetilde{\mathbf{r}}. \tag{26}$$

The regularized system is solved by direct factorization, with rank-revealing QR used for numerically ill-conditioned cases.

The complete forecast restores the residual to the original traffic-flow scale and adds it to the anchor:

$$\widehat{y}_{t+h} = b_t^{(\alpha)} + \mu_r + \sigma_r \mathbf{z}_t^\top \widehat{\boldsymbol{\beta}}. \tag{27}$$

2.4. Model Selection and Forecasting

The shared fractional-prior parameters α and λ_A, together with lag length P, are determined from the development data before outer-test evaluation. With these quantities fixed, an ELM configuration

$$\boldsymbol{\theta} = (L, \rho, \lambda_R) \in \Theta_R$$

is selected for each station i. The configuration specifies the hidden-layer width, activation function, and readout regularization. Candidate ranges, fold counts, and selected values are reported in the experimental settings.

Model selection uses temporally ordered rolling folds with expanding training windows. Samples entering training, validation, and testing contain a complete lag block and all required seasonal observations. Let $\mathcal{T}_{i,f}$ and $\mathcal{V}_{i,f}$ denote the training and validation samples in fold f. The anchor is fitted on $\mathcal{T}_{i,f}$, and the resulting training residuals define the residual standardization statistics. Input standardization parameters are estimated from the same training samples. Each candidate ELM is fitted on $\mathcal{T}_{i,f}$ and evaluated on $\mathcal{V}_{i,f}$ using the complete forecast in Equation (27). The selected configuration minimizes mean validation MAE across folds:

$$\boldsymbol{\theta}_i^{\star} = \arg\min_{\boldsymbol{\theta}\in\Theta_R} \frac{1}{F}\sum_{f=1}^{F} \frac{1}{\left|\mathcal{V}_{i,f}\right|} \sum_{t\in\mathcal{V}_{i,f}} \left| y_{i,t+h} - \widehat{y}_{i,t+h}^{(f)}(\boldsymbol{\theta}) \right|. \tag{28}$$

After model selection, the anchor is re-estimated on the complete development set. Development residuals and standardization statistics are then recomputed, and the final ELM readout is fitted using $\boldsymbol{\theta}_i^{\star}$. The resulting model generates forecasts for the outer test set, which are evaluated using MAE, RMSE, and MAPE. Algorithm 1 summarizes the complete procedure.

Algorithm 1 Training and forecasting procedure of FPA-ELM.

Require: Development sets $\{\mathcal{D}_{i,\text{dev}}\}_{i=1}^{I}$; outer-test sets $\{\mathcal{D}_{i,\text{test}}\}_{i=1}^{I}$; forecasting horizon h; input length P; seasonal period S; seasonal order J; fractional-prior parameters α and λ_A; ELM candidate set Θ_R; number of rolling folds F; random seed

Ensure: Station-specific FPA-ELM models and outer-test forecasts

Stage 1: Fractional-prior construction

Derive the fractional-prior weights $\mathbf{w}_\alpha$ using Equations (5) and (6) and form the penalty matrix $\mathbf{P}_\alpha$
for each station $i = 1, \ldots, I$ **do**
    Form development samples from lagged observations and seasonal information

Stage 2: Rolling validation and model selection

    for each rolling fold $f = 1, \ldots, F$ **do**
        Split the development samples into the temporally ordered training set $\mathcal{T}_{i,f}$ and validation set $\mathcal{V}_{i,f}$
        Estimate the fractional-prior anchor on $\mathcal{T}_{i,f}$ using Equation (11)
        Compute the residual target

$$r_{i,t} = y_{i,t+h} - b_{i,t}^{(\alpha)}$$

        Standardize the model inputs and residual targets using statistics from $\mathcal{T}_{i,f}$
        for each candidate $\boldsymbol{\theta} \in \Theta_R$ **do**
            Construct the fixed random ELM representation specified by $\boldsymbol{\theta}$
            Estimate the residual readout using Equation (26)
            Combine the anchor and residual correction to predict $\mathcal{V}_{i,f}$
            Compute the validation error $E_{i,f}(\boldsymbol{\theta})$
        end for
    end for
    Select the station-specific configuration $\boldsymbol{\theta}_i^{\star}$ using Equation (28)

Algorithm 1 *Cont.*

Stage 3: Final model estimation
Re-estimate the fractional-prior anchor on the complete development set $\mathcal{D}_{i,\mathrm{dev}}$
Recompute the residual targets and standardization parameters
Fit the final ELM residual model using $\boldsymbol{\theta}_i^{\star}$

Stage 4: Outer-test forecasting
for each admissible outer-test origin t **do**
Compute the anchor prediction $b_{i,t}^{(\alpha)}$
Compute the ELM residual correction $\widehat{r}_{i,t}$
Form the final forecast

$$\widehat{y}_{i,t+h} = b_{i,t}^{(\alpha)} + \widehat{r}_{i,t}$$

end for
end for
Evaluate the outer-test forecasts using MAE, RMSE, and MAPE
return fitted FPA-ELM models and outer-test forecasts

3. Experiment and Analysis

3.1. Experimental Settings

3.1.1. Dataset Description

The experiments used 36 station-level univariate traffic-flow series from the United Kingdom and the United States, including 12 series from the United Kingdom (UK), 6 from Maryland (MD), 12 from Minnesota (MN), and 6 from Washington (WA). These series correspond to 30 unique physical monitoring stations; the MN subset contains two annual records for each of six stations, yielding 12 station-year series. The corresponding highways, source station IDs, and lane counts are listed in Table 1.

Traffic flow was recorded at 15-min intervals, corresponding to 96 observations per day and 672 observations per week. Individual series contain 8736–35,136 observations, covering approximately 91–366 days. Continuous zero padding at the beginning or end of each series was removed before sample construction, while internal zero-flow observations were retained. The resulting dataset contains 852,946 observations with no missing values. Traffic flow ranges from 0 to 2389.33 veh/15 min.

Table 1. Traffic-flow series used in the experiments.

Region	Highway	Source Station ID	Lanes
UK	M25	4762a	4
UK	M25	4762b	4
UK	M25	4822a	4
UK	M25	4826a	4
UK	M25	4868a	4
UK	M25	4868b	4
UK	M25	4565a	4
UK	M25	4680b	4
UK	M1	2737a	3
UK	M1	2808b	3
UK	M1	4897a	3
UK	M6	6951a	3

Table 1. *Cont.*

Region	Highway	Source Station ID	Lanes
MD	I270	2a	3
MD	I95	4b	4
MD	I795	7a	2
MD	I795	7b	2
MD	I695	9a	4
MD	I695	9b	4
MN	I35W-NB	60	4
MN	I35W-SB	578	3
MN	I35E-NB	882	3
MN	I35E-SB	890	3
MN	169-NB	442	2
MN	169-SB	737	2
MN	I35W-NB	60	4
MN	I35W-SB	578	3
MN	I35E-NB	882	3
MN	I35E-SB	890	3
MN	169-NB	442	2
MN	169-SB	737	2
WA	I5	ES-179D_MN_Stn	4
WA	I5	ES-179D_MS_Stn	3
WA	I5	ES-130D_MN_Stn	4
WA	I5	ES-130D_MS_Stn	4
WA	I405	ES-738D_MN_Stn	3
WA	I405	ES-738D_MS_Stn	3

3.1.2. Forecasting Task

The primary task is one-step traffic-flow forecasting with $h = 1$, corresponding to a 15-min-ahead prediction. Additional experiments use $h \in \{4, 8, 96\}$, corresponding to horizons of 1 h, 2 h, and 24 h. Each traffic-flow series is modeled independently.

The main FPA-ELM experiments use $(W_l, W_c) = (12, 12)$. The lag window W_l supplies recent observations to the residual readout, while the context window W_c determines the observations used to calculate the local standard deviation, minimum, and maximum. The residual input also contains target-time calendar features defined in Section 2.3. The anchor uses the current observation together with $J = 7$ same-phase seasonal observations.

All baseline methods used the same 12-step historical traffic-flow window as the lag input of FPA-ELM and generated scalar point forecasts. The input observations were drawn from the same original station-level series, without incorporating neighboring-station measurements or road-network topology. FPA-ELM additionally constructed its proposed seasonal anchor and calendar-based features from the available historical information.

3.1.3. Chronological Selection and Fitting

Each trimmed series was split chronologically into an 80% development period and a 20% outer-test period. For a sequence of length n, the outer-test period begins at target index $\lfloor 0.8n \rfloor$. Temporal order was preserved throughout model selection and evaluation.

FPA-ELM used four-fold rolling-origin cross-validation with expanding training windows within the development period. Each validation block contained at least 192 samples and approximately 5% of the usable development window, while the initial training block covered at least 45% of the development period. The weekday offset for the calendar representation was estimated before the first validation block and then kept fixed.

For each station, the ELM readout configuration $\boldsymbol{\theta} = (L, \rho, \lambda_R)$ was selected by the mean validation MAE across the four folds. The fractional-prior parameters (α, λ_A) were specified as horizon-dependent group constants shared across stations. After model selection, the anchor, standardization parameters, and residual readout were refitted on the complete development period before outer-test forecasting. The selected configurations were subsequently refitted on the complete development period and evaluated on the same outer-test windows. Three random seeds were used for stochastic components.

3.1.4. Optimization Settings

FPA-ELM is estimated through regularized linear systems. The fractional-prior anchor uses penalized least squares with a diagonal jitter of 10^{-10}, while the ELM residual readout uses ridge-regularized least squares with an unpenalized intercept. Rank-revealing QR implemented through `gelsy` is used for final readout estimation. Gram-matrix caching and Cholesky factorization accelerate repeated readout estimation during cross-validation. The random hidden-layer weights and biases are generated from fixed random streams and remain unchanged during readout fitting.

The conventional baselines use fixed configurations or compact validation grids. Deep-learning baselines are trained with Adam and MSE loss using model-specific batch sizes, epoch budgets, and early-stopping settings. Detailed parameter settings are summarized in Table 2.

Table 2. Hyperparameter settings of FPA-ELM and baseline methods.

Method	Parameter Settings
FPA-ELM anchor	$\alpha \in \{0.2, 0.5, 0.8, 1.0, 1.5\}$; $\lambda_A \in \{0.01, 0.1, 1.0\}$; $J = 7$
FPA-ELM readout	$L \in \{200, 400\}$; $\rho \in \{\tanh, \mathrm{ReLU}\}$; $\lambda_R \in \{10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4\}$
Input windows	Main setting: $(W_l, W_c) = (12, 12)$; sensitivity grid: $W_l, W_c \in \{1, 2, 3, 6, 12, 24\}$

3.1.5. Evaluation Metrics

Forecasting accuracy is evaluated using mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE). For M test targets, the metrics are

$$\begin{aligned} \mathrm{MAE} &= \frac{1}{M}\sum_{m=1}^{M} |y_m - \widehat{y}_m|, \\ \mathrm{RMSE} &= \sqrt{\frac{1}{M}\sum_{m=1}^{M} (y_m - \widehat{y}_m)^2}, \\ \mathrm{MAPE} &= \frac{100\%}{|\mathcal{I}_\delta|}\sum_{m\in\mathcal{I}_\delta}\left|\frac{y_m - \widehat{y}_m}{y_m}\right|, \qquad \mathcal{I}_\delta = \{m : |y_m| > \delta\}. \end{aligned} \tag{29}$$

Here, $\delta = 10^{-8}$. Zero-flow targets are excluded from MAPE and retained in MAE and RMSE.

Metrics are averaged across the three random seeds within each series and then equally across the 36 series. Paired comparisons use two-sided Wilcoxon signed-rank tests, with Holm adjustment for multiple comparisons within each family of tests. Matched-pairs rank-biserial correlation, r_{rb}, is reported as the effect size, and 95% bootstrap confidence intervals are used to quantify uncertainty in paired performance differences.

Because the MN subset contains repeated annual records from the same six physical stations, a sensitivity analysis is also performed at the physical-station level. For each repeated MN station, the paired differences from the two annual series are averaged,

yielding 30 physical-station-level observations. The Wilcoxon tests are repeated on these 30 observations, and bootstrap resampling is conducted at the physical-station level.

3.1.6. Hyperparameter Ranges

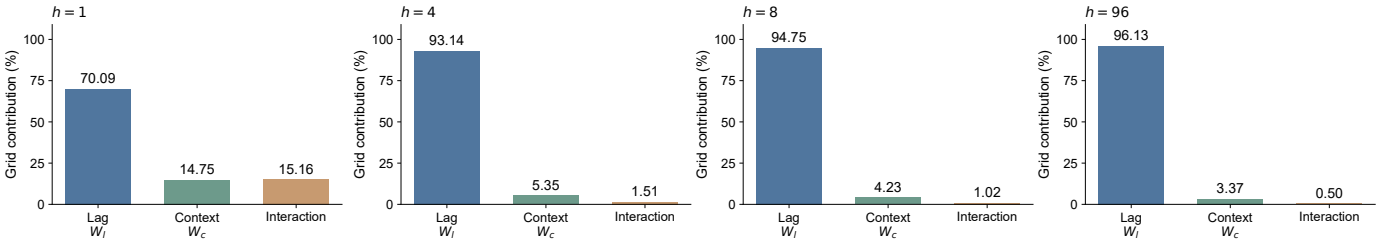
The fractional-prior parameters were selected on 12 calibration stations and then shared across stations at each forecasting horizon. The resulting settings were $(\alpha, \lambda_A) = (0.5, 0.1)$ for $h = 1$, $(0.8, 0.1)$ for $h = 4$ and $h = 8$, and $(0.2, 0.1)$ for $h = 96$. The seasonal-pool depth was fixed at $J = 7$.

The ELM readout configuration was selected independently for each station using four-fold rolling-origin cross-validation. The main experiments used $(W_l, W_c) = (12, 12)$, while the sensitivity analysis examined alternative lag and context widths. Candidate ranges for FPA-ELM are summarized in Table 2.

3.2. Sensitivity Analysis of the Input Windows

FPA-ELM uses two temporal windows. The lag window W_l supplies recent observations to the residual readout. The context window W_c provides the observations used to calculate the local standard deviation, minimum, and maximum in Equation (16). The default setting $(W_l, W_c) = (12, 12)$ covers three hours at 15-min intervals. We evaluated all 36 pairs from $W_l, W_c \in \{1, 2, 3, 6, 12, 24\}$ at 36 stations, using three random seeds and horizons $h \in \{1, 4, 8, 96\}$. Anchor inputs, (α, λ_A), data splits, and sample ranges remained fixed, while the readout configuration was re-selected for each pair. We averaged the three seeds within each station and then weighted the 36 stations equally. Factor contributions were estimated by orthogonal sum-of-squares decomposition of the 6×6 grid means.

Lag length dominated window sensitivity, with a larger contribution at longer horizons (Figure 2). At $h = 1$, W_l, W_c, and their interaction accounted for 70.1%, 14.7%, and 15.2% of the grid variation. The contribution of W_l rose to 93.1%, 94.7%, and 96.1% at $h = 4$, 8, and 96, respectively. Over the same horizons, the contribution of W_c fell to 5.4%, 4.2%, and 3.4%, while the interaction fell to 1.5%, 1.0%, and 0.5%.



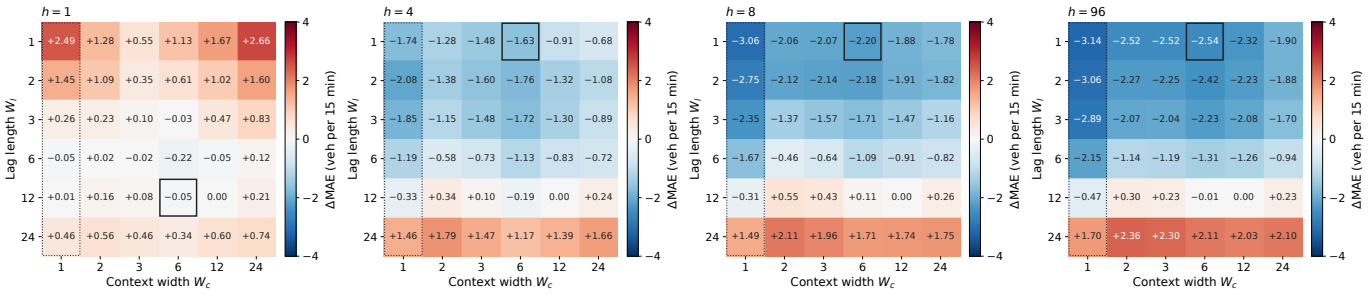
(**a**) $h = 1$ (**b**) $h = 4$ (**c**) $h = 8$ (**d**) $h = 96$

Figure 2. Contributions of lag length, context width, and their interaction to MAE variation across forecasting horizons.

The response surfaces show that the preferred lag length also shortened with the forecasting horizon (Figure 3). Relative to $(12, 12)$, the marginal MAE changes for $W_l = 1$ and 2 were -2.18 and -2.15 at $h = 8$. They were -2.49 and -2.35 at $h = 96$. By contrast, $W_l = 24$ increased MAE by 1.79 and 2.10 at these horizons.

This pattern is consistent with the temporal roles of the anchor and residual readout. The anchor represents the latest flow level and same-phase seasonal history, leaving the readout to model departures from that reference. At longer horizons, additional recent lags may duplicate short-term persistence already represented by the anchor. Compact lag inputs therefore gave lower MAE in these experiments.

The one-step results differed. At $h = 1$, reducing W_l to one increased marginal MAE by 1.63, and the W_l–W_c interaction explained 15.2% of grid variation. Multiple recent observations retain local trajectory information that the current value alone does not provide. Accordingly, the preferred lag length changed from $W_l = 12$ at $h = 1$ to $W_l = 1$ at $h \geq 4$.



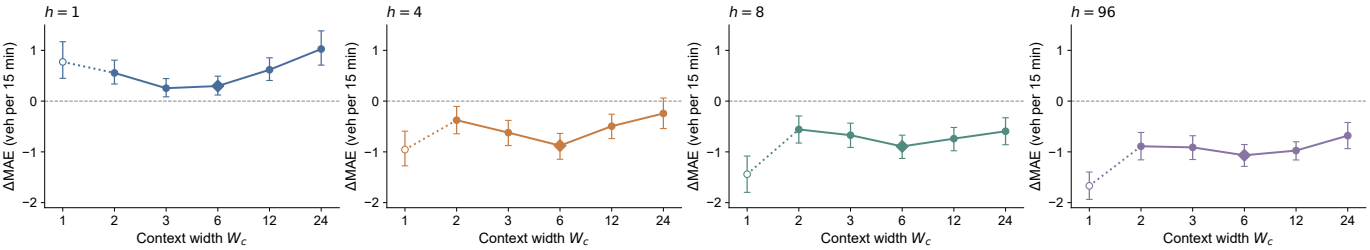
(**a**) $h = 1$ (**b**) $h = 4$ (**c**) $h = 8$ (**d**) $h = 96$

Figure 3. MAE changes over the lag–context grid relative to $(12, 12)$, with negative values indicating improvement and solid outlines marking recommended configurations.

Context width had a smaller effect (Figure 4). At $h = 1$, $W_c = 3$ had the lowest marginal MAE among the tested widths, with a change of $+0.25$ relative to $(12, 12)$. At $h \geq 4$, the numerical minimum occurred at $W_c = 1$. In this degenerate setting, the local standard deviation is zero and both extrema equal y_t. Among non-degenerate settings, $W_c = 6$ gave the lowest marginal MAE at $h = 4$, 8, and 96, with changes of -0.88, -0.89, and -1.07.

The stronger interaction at $h = 1$ made the preferred context width depend more on lag length. At $W_l = 12$, changing W_c from 12 to 6 reduced MAE from 44.478 to 44.432. This local improvement was not captured by the positive marginal effect of $W_c = 6$ averaged across all lag lengths.

The two windows serve different functions. The lag window supplies temporal predictors, whereas the context window sets the scale for summarizing local variability. Very short contexts contain little variation, while long contexts weaken locality. The results favored $W_c = 6$, a 90-min window, among the non-degenerate settings.



(**a**) $h = 1$ (**b**) $h = 4$ (**c**) $h = 8$ (**d**) $h = 96$

Figure 4. Marginal MAE effects of context width relative to $(12, 12)$ across forecasting horizons, averaged over lag lengths with 95% station-level bootstrap intervals.

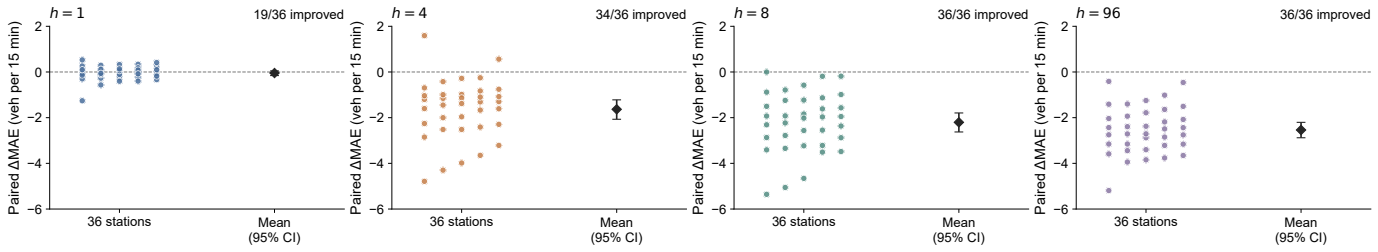
The selection rule retained non-degenerate cells within 0.25 MAE of the best eligible configuration. It then prioritized smaller W_l and W_c, in that order. The rule selected $(3, 6)$ at $h = 1$, but the recommended setting retained $W_l = 12$ because the MAE gain was below the 0.5 adoption threshold. At $h = 4$, both the rule and final recommendation selected $(1, 6)$. At $h = 8$ and 96, the rule selected $(1, 2)$, while $(1, 6)$ achieved slightly lower mean

MAE and used the same non-degenerate context width across longer horizons. The final settings were therefore $(12, 6)$ at $h = 1$ and $(1, 6)$ at $h \geq 4$ (Table 3).

Table 3. Selected input windows and paired MAE changes relative to $(12, 12)$, with p_{FDR} denoting Benjamini–Hochberg-adjusted paired Wilcoxon p values, † marking deviations from the selection rule, and dashes indicating unavailable confirmations.

h	Rule (W_l, W_c)	Recommended (W_l, W_c)	MAE (12, 12)	MAE Recommended	ΔMAE	Stations Improved	p_{FDR}	Arm 1 ΔMAE	Arm 2 ΔMAE
1	$(3, 6)$	$(12, 6)$ †	44.478	44.432	−0.05	19/36	0.63	—	—
4	$(1, 6)$	$(1, 6)$	60.864	59.230	−1.63	34/36	4.8×10^{-8}	—	—
8	$(1, 2)$	$(1, 6)$ †	69.302	67.098	−2.20	36/36	8.7×10^{-10}	−2.22	−2.02
96	$(1, 2)$	$(1, 6)$ †	72.449	69.910	−2.54	36/36	7.9×10^{-11}	−2.09	−2.56

The station-level paired MAE differences for the recommended input windows are shown in Figure 5.



(**a**) $h = 1$ (**b**) $h = 4$ (**c**) $h = 8$ (**d**) $h = 96$

Figure 5. Station-level three-seed mean MAE differences between recommended and default $(12, 12)$ settings, with negative values indicating improvement and black diamonds showing the mean with 95% bootstrap intervals.

Relative to $(12, 12)$, the recommended settings reduced test MAE by 1.63, 2.20, and 2.54 veh per 15 min at $h = 4$, 8, and 96. MAE decreased at 34 of 36 stations at $h = 4$ and at all 36 stations at $h = 8$ and 96. All three paired differences remained significant after Benjamini–Hochberg correction. At $h = 1$, changing W_c from 12 to 6 reduced mean MAE by 0.05, with improvements at 19 of 36 stations and no significant paired difference.

The confirmation experiments reproduced the long-horizon results. Across the two arms, $(1, 6)$ reduced MAE by 2.22 and 2.02 at $h = 8$, and by 2.09 and 2.56 at $h = 96$. These results support a common non-degenerate context width of $W_c = 6$, with $W_l = 12$ for one-step forecasting and $W_l = 1$ for horizons of at least one hour.

3.3. Ablation Analysis

We evaluated four configurations to isolate the contributions of the anchor, residual readout, and direct link. These were anchor only, readout only with the direct link, FPA-ELM without the direct link, and complete FPA-ELM. Experiments used one-step forecasts at 36 traffic stations, the same 80/20 outer split, and three random seeds. Each ELM-based configuration was tuned independently by four-fold rolling-origin cross-validation over the same 28 candidates. Errors were calculated by station and then averaged across stations. MAPE is reported in percent, while MAE and RMSE are in vehicles per 15 min. Zero-flow samples were excluded from MAPE, and all differences were computed from unrounded results.

Complete FPA-ELM achieved the lowest mean RMSE, MAE, and MAPE among the four configurations (Table 4).

Removing the residual readout produced the largest degradation. Relative to complete FPA-ELM, the anchor-only model increased mean RMSE by 17.49, MAE by 13.20, and MAPE by 2.26 percentage points. The anchor represents persistent and same-phase seasonal structure, while the readout models departures from this reference. The large error increase therefore shows that residual correction contributed most of the one-step accuracy gain.

Table 4. Mean forecasting errors of the ablation variants.

Algorithm	Mean RMSE	Mean MAE	Mean MAPE (%)
Readout only (with direct link)	67.59	45.59	9.13
Anchor only	83.92	57.68	11.18
FPA-ELM without direct link	66.49	44.56	8.93
FPA-ELM	**66.42**	**44.48**	**8.92**

Note: Bold values indicate the lowest forecasting errors for each metric.

Removing the anchor had a smaller but consistent effect. Complete FPA-ELM reduced RMSE, MAE, and MAPE by 1.73%, 2.43%, and 2.30% relative to the readout-only model. The lag inputs already contain a strong persistence signal, allowing the readout-only model to recover much of the next-step flow. The anchor adds an explicit baseline and same-phase information that is absent from the ELM input vector, reducing the residual burden on the readout.

Paired comparisons used the three-seed mean at each station, giving 36 observations per comparison. Two-sided Wilcoxon signed-rank tests were adjusted jointly across the nine variant–metric comparisons using the Holm procedure. Here, Δ is the mean station-level error of the ablation variant minus that of FPA-ELM. The matched-pairs rank-biserial correlation is defined as $r_{\mathrm{rb}} = (R_+ - R_-)/(R_+ + R_-)$, where R_+ and R_- denote the sums of positive and negative signed ranks after zero differences are excluded. It is reported alongside the standardized mean paired difference d_z. W/T/L counts stations with lower, equal, or higher ablation error. The paired statistical comparisons are reported in Table 5.

Table 5. Paired error comparisons between FPA-ELM and its ablation variants.

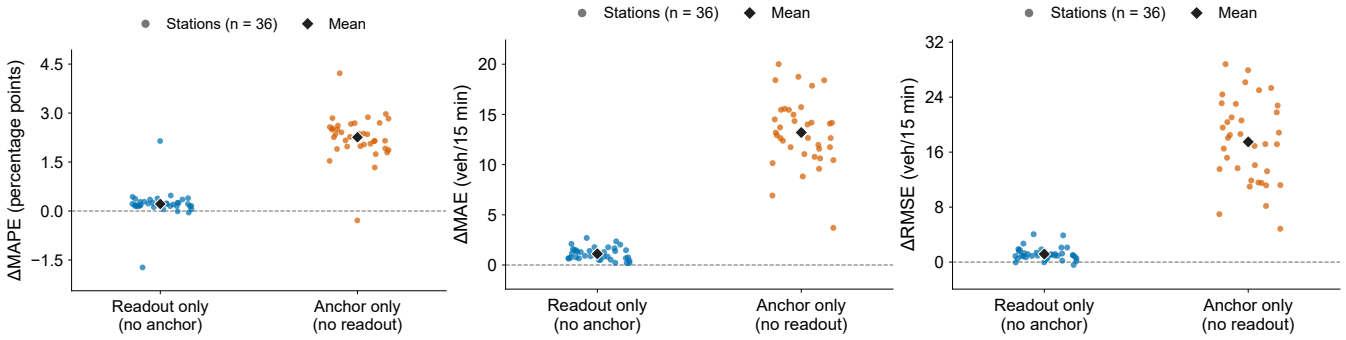
Algorithm	Metric	Δ	p_{Holm}	r_{rb}	d_z	W/T/L
Readout only (with direct link)	RMSE	+1.17	4.8×10^{-9}	+0.973	+1.23	3/0/33
	MAE	+1.11	2.6×10^{-10}	+1.000	+1.80	0/0/36
	MAPE	+0.22	6.6×10^{-7}	+0.883	+0.45	3/0/33
Anchor only	RMSE	+17.49	2.6×10^{-10}	+1.000	+2.90	0/0/36
	MAE	+13.20	2.6×10^{-10}	+1.000	+3.95	0/0/36
	MAPE	+2.26	3.5×10^{-10}	+0.997	+3.40	1/0/35
FPA-ELM without direct link	RMSE	+0.06	0.0044	+0.535	+0.33	9/0/27
	MAE	+0.08	4.3×10^{-7}	+0.898	+1.12	2/0/34
	MAPE	+0.02	7.3×10^{-5}	+0.742	+0.32	4/0/32

MAPE differences are expressed in percentage points. All statistics use unrounded station-level means over three seeds ($n = 36$), with Holm correction across all nine tests. Positive Δ and r_{rb} values favor complete FPA-ELM. The p_{Holm} and d_z values are retained from the original analysis; r_{rb} is newly reported.

FPA-ELM had lower MAE than either single-component model at all 36 stations. The anchor-only model (without the residual readout) increased mean MAE by 13.20, with station-level increases of 3.70–20.01. In contrast, the readout-only model (without the anchor) increased mean MAE by 1.11, with station-level increases of 0.19–2.69. Both comparisons remained significant after Holm adjustment. For MAPE, the anchor-only model performed better at one station and the readout-only model at three stations (Figure 6).

These results indicate that both components contributed across stations, with a larger contribution from the residual readout.

The fitted anchor assigned mean coefficients of 0.82 to the latest observation and 0.18 to the same-phase seasonal pool (Figure 7). Thus, the anchor remained close to persistence while incorporating a smaller seasonal adjustment. Relative to persistence, it reduced MAE and RMSE at all 36 stations (Figure 8). Mean station-level reductions were 6.42% for MAE and 7.02% for RMSE, and median MAE decreased from 61.09 to 57.63.



(**a**) MAPE (**b**) MAE (**c**) RMSE

Figure 6. Station-level error differences for the single-component ablations.

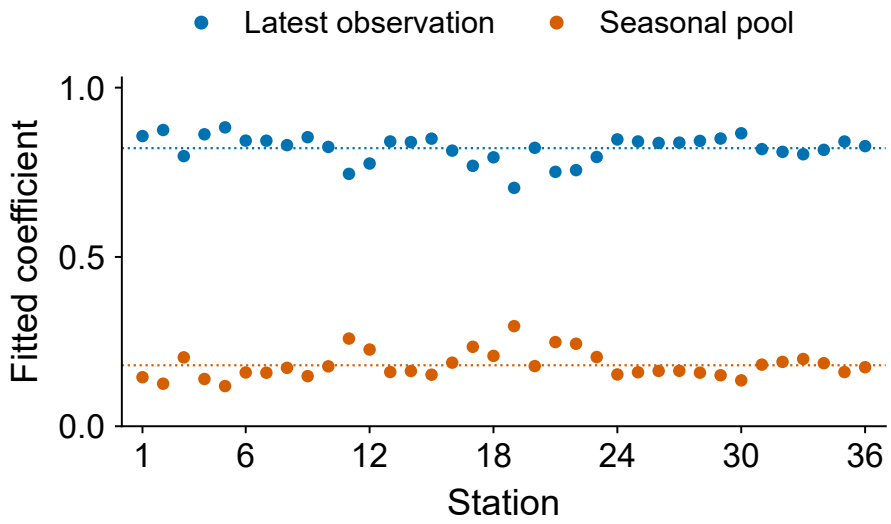


Figure 7. Fitted anchor coefficients across stations.

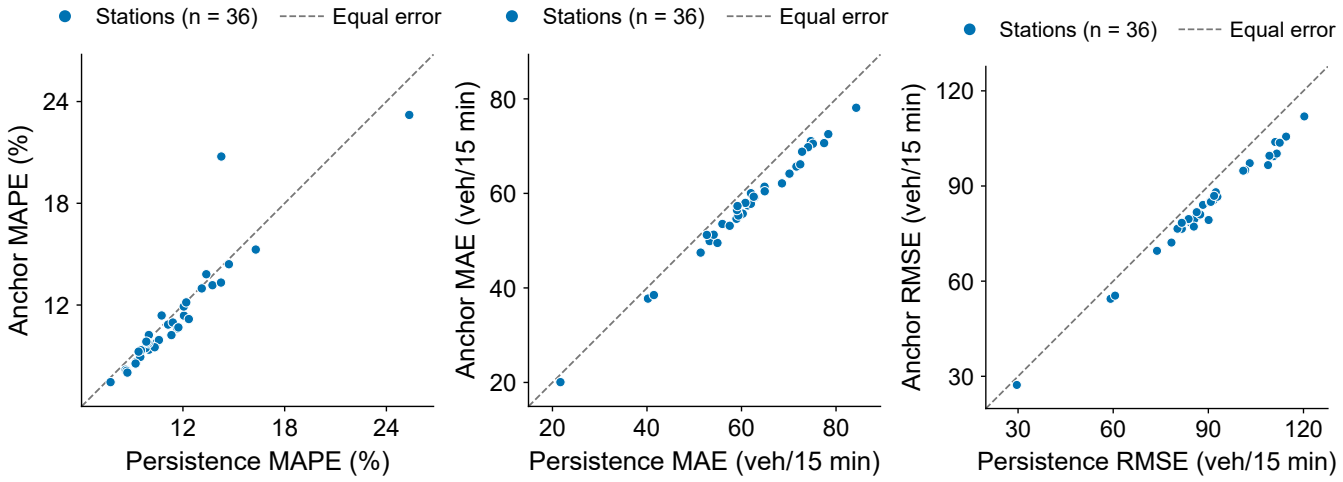
MAPE decreased at 31 stations, with a mean relative reduction of 2.82%, but increased at five stations. At station 27, MAPE rose from 14.26% to 20.76%, a relative increase of 45.54%. MAPE weights errors by observed flow and is therefore more sensitive than MAE or RMSE to deviations during low-flow periods.

The seasonal coefficient was positively associated with relative reductions in MAPE, MAE, and RMSE, with Spearman correlations of 0.34, 0.61, and 0.76 (Figure 9). The stronger MAE and RMSE associations are consistent with greater anchor gains at stations receiving more seasonal weight, but the correlations remain descriptive. Relative reduction was calculated as $100(E_{\text{persistence}} - E_{\text{anchor}})/E_{\text{persistence}}$; negative values indicate higher anchor error. Table 6 summarizes the station-level anchor and persistence results.

The direct link yielded smaller gains than the anchor or residual readout. Removing it increased mean MAE from 44.48 to 44.56, with higher MAE at 34 of 36 stations (Figure 10b). Retaining the link reduced mean MAE by 0.18% relative to the no-link model. MAPE was

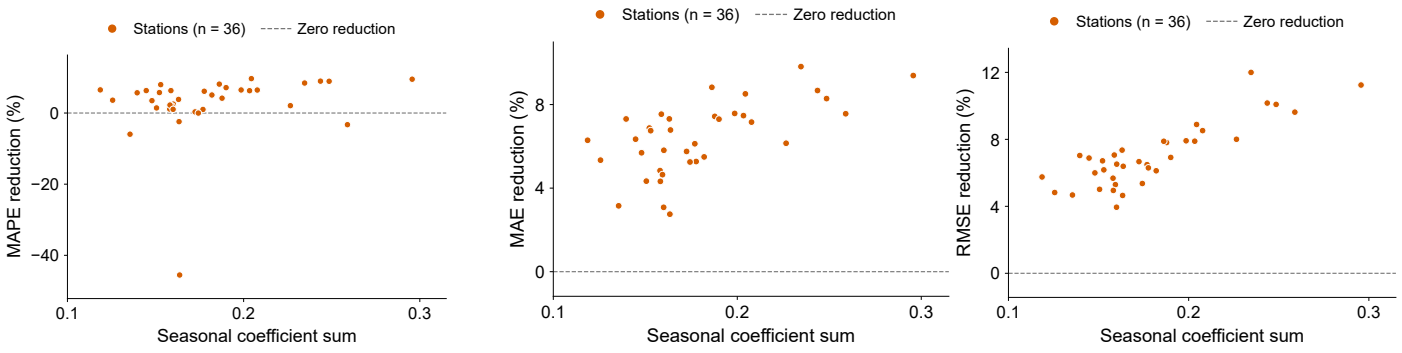
also higher without the link at 32 stations and lower at four. The Holm-adjusted differences were significant but small.

Mean RMSE increased from 66.42 to 66.49 without the link, with $d_z = 0.33$. RMSE increased at 27 stations and decreased at nine; the largest reported exception was station 31, where removing the link reduced RMSE by 0.63. The direct link provides a linear route from the residual inputs to the output alongside the nonlinear random-feature branch. The small and less uniform RMSE effect suggests that this linear correction mainly affected routine residual variation. Complete and no-link models selected the same (L, ρ, λ_R) in all 108 station–seed runs, isolating the comparison from changes in the selected readout configuration.



(**a**) MAPE (**b**) MAE (**c**) RMSE

Figure 8. Station-level errors of the anchor and persistence models.

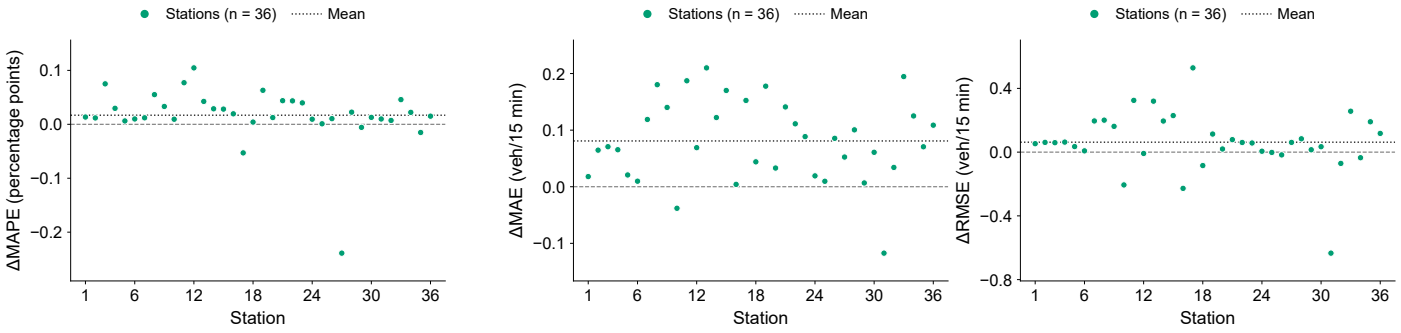


(**a**) MAPE (**b**) MAE (**c**) RMSE

Figure 9. Seasonal coefficients and relative error reductions.

Table 6. Station-level anchor performance versus persistence and its association with the seasonal coefficient.

Metric	Anchor Mean	Persistence Mean	Mean Absolute Reduction	Mean Relative Reduction (%)	Stations Improved	Relative-Reduction Range (%)	ρ_s
MAPE	11.18	11.48	0.30	2.82	31/36	−45.54–9.67	0.34
MAE	57.68	61.66	3.98	6.42	36/36	2.75–9.82	0.61
RMSE	83.92	90.31	6.39	7.02	36/36	3.95–12.00	0.76



(**a**) MAPE (**b**) MAE (**c**) RMSE

Figure 10. Station-level error differences after removing the direct link.

3.4. Evaluation of the Fractional Seasonal Prior

Three seasonal priors were tested under the same FPA-ELM configuration: fractional, uniform, and exponential. Only the prior profile was changed, and all variants were evaluated on the same 36 station-level series using MAE, RMSE, and MAPE.

The fractional profile gave the lowest average error for all three metrics (Table 7), and the station-level changes are shown in Figure 11. Compared with uniform weights, it reduced MAE by 2.27%, RMSE by 2.32%, and MAPE by 3.31%, from 9.22% to 8.92%. It produced lower values for all three metrics at every station. Against the exponential profile, mean MAE, RMSE, and MAPE were lower by 1.17%, 1.47%, and 1.70%, respectively. The fractional profile had lower MAE and RMSE at all 36 stations and lower MAPE at 35 stations. These reductions were small but consistent across stations and support the chosen profile for FPA-ELM.

Table 7. Mean errors across 36 stations for three seasonal priors, with lower values indicating better forecasts.

Prior Profile	MAE	RMSE	MAPE (%)
Uniform	45.51	68.00	9.22
Fractional	**44.48**	**66.42**	**8.92**
Exponential	45.00	67.41	9.07

Note: Bold values indicate the lowest forecasting errors for each metric.

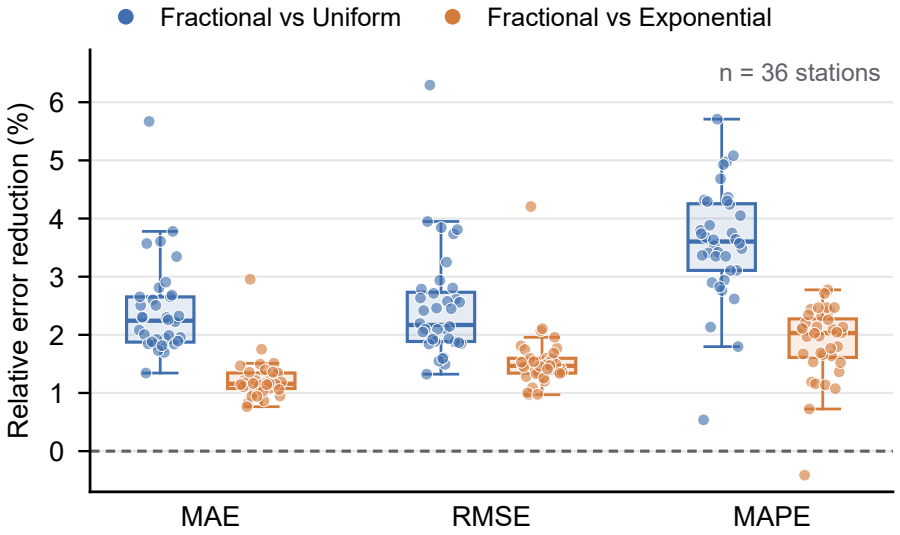


Figure 11. Station-level relative error reductions of the fractional prior against uniform and exponential profiles.

3.5. Traffic Flow Prediction Under Different Methods

To assess the forecasting performance of FPA-ELM across different modeling paradigms, 21 representative baseline methods were selected, covering conventional statistical forecasting, classical machine learning, ensemble learning, and deep-learning approaches. The selection includes methods based on linear temporal modeling, nonlinear regression, tree-based ensembles, recurrent networks, convolutional architectures, and attention-based sequence modeling. This diversity enables the proposed method to be evaluated against approaches with different assumptions and representation capabilities, rather than relying solely on the number of baselines.

The conventional baselines included ARIMA, multilayer perceptron (MLP) [22], *K*-nearest neighbors (KNN) [23], decision tree (DT) [24], support vector regression (SVR) [25], random forest (RF) [26], XGBoost [27], CatBoost [28], and LightGBM [29]. The deep-learning baselines comprised BiTCN, DeepAR [30], SegRNN [31], DLinear [32], N-BEATS [33], Mamba [34], Transformer [6], PatchTST [35], LightTS [36], Informer [37], iTransformer [38], and Autoformer [39].

BiTCN encoded the historical window in forward and reversed order using two temporal-convolution branches [40]. The resulting representations were concatenated and mapped to a scalar forecast through a linear output layer. The selected configuration used channel sizes $[16, 16]$, a kernel size of 3, a dropout rate of 0.1, a learning rate of 0.001, and a batch size of 128. DeepAR retained the LSTM-based autoregressive backbone and used a scalar decoder trained with mean squared error for point forecasting. The remaining deep-learning models were evaluated under the same one-step station-level setting.

All methods were evaluated using the same chronological data partitions, 12-step historical traffic-flow windows, one-step forecasting horizon, and evaluation metrics. These consistent experimental conditions reduce potential biases arising from differences in data partitioning, input window length, and parameter selection. Nevertheless, architectural differences and model-specific optimization procedures remain inherent to comparisons across different forecasting paradigms. For the baseline methods, the final 10% of the development period was reserved as a chronological validation segment. Model-specific hyperparameters were selected using validation performance where tuning was performed, and the validation segment was also used for early stopping when applicable.

Table 8 reports the mean errors across the 36 traffic-flow series, representing 30 unique physical monitoring stations. FPA-ELM achieved the lowest values for all three metrics, with an MAE of 44.48, RMSE of 66.42, and MAPE of 8.92%.

MAE and RMSE are expressed in vehicles per 15 min, while MAPE is expressed in percent. Each value in Table 8 is the arithmetic mean of the corresponding station-level errors across the 36 series, with the three-seed mean computed within each series before equal-weight aggregation.

Among the conventional methods, XGBoost achieved the lowest MAE and RMSE, at 47.06 and 69.62, while RF achieved the lowest MAPE of 9.14%. Relative to these strongest conventional results, FPA-ELM reduced MAE by 5.49%, RMSE by 4.59%, and MAPE by 2.41%. CatBoost also produced low MAE and RMSE values of 47.42 and 69.83, but remained above FPA-ELM on all three metrics.

N-BEATS was the strongest deep-learning baseline, with an MAE of 48.58, RMSE of 70.98, and MAPE of 9.61%. FPA-ELM reduced these errors by 8.44%, 6.42%, and 7.18%, respectively. Transformer, Informer, Autoformer, Mamba, and iTransformer produced larger one-step errors despite their greater sequence-modeling capacity. For this short-horizon task, additional sequence complexity did not translate into lower forecasting error.

Figure 12a–c compares the relative improvement of FPA-ELM over all 21 baselines. For method m, the reduction is defined as

$$R_m = 100 \frac{\bar{E}_m - \bar{E}_{\text{FPA-ELM}}}{\bar{E}_m},$$

where $\bar{E}_m$ denotes the arithmetic mean error across the 36 series. Positive values indicate lower error with FPA-ELM. The improvements are largest against methods with higher one-step errors, while the margins narrow against the strongest tree-based and deep-learning baselines.

Table 8. One-step forecasting performance of FPA-ELM and baseline methods.

Algorithm	MAE	RMSE	MAPE (%)
Conventional methods			
ARIMA	57.67	85.04	10.69
MLP	48.98	71.37	9.84
KNN	49.81	74.69	9.59
DT	53.65	79.41	10.22
SVR	47.88	70.94	9.41
RF	47.46	70.69	9.14
XGBoost	47.06	69.62	9.30
CatBoost	47.42	69.83	9.36
LightGBM	48.36	71.75	9.17
Deep-learning methods			
BiTCN	55.24	76.86	14.35
DeepAR	50.44	73.11	10.26
SegRNN	49.38	71.84	9.84
DLinear	59.62	85.34	13.56
N-BEATS	48.58	70.98	9.61
Mamba	58.95	83.03	12.46
Transformer	51.30	74.17	10.63
PatchTST	49.37	71.48	10.29
LightTS	49.34	71.79	9.94
Informer	52.28	74.97	10.62
iTransformer	51.64	73.88	11.08
Autoformer	61.72	88.70	13.41
FPA-ELM	**44.48**	**66.42**	**8.92**

Note: Bold values indicate the lowest forecasting errors for each metric.

For statistical inference, the three-seed mean error was computed for each of 36 series, yielding 36 paired differences for each baseline comparison. One-sided Wilcoxon signed-rank tests assessed the directional hypothesis of lower error with FPA-ELM, and the resulting p-values were adjusted by the Holm procedure separately for MAE, RMSE, and MAPE. Among the 21 baseline methods reported in Table 8, FPA-ELM achieved significantly lower MAE and RMSE than all 21 methods after correction ($p_{\text{Holm}} < 0.05$ for each comparison). For MAPE, 19 of the 21 comparisons remained significant. The differences against RF and LightGBM were not significant either before or after adjustment (one-sided raw $p = 0.061$ and $p_{\text{Holm}} = 0.123$ for both).

The series-level differences in Figure 13a–c provide a direct view of the improvements over XGBoost, RF, and N-BEATS. For series i and baseline m, the paired difference is

$$\Delta E_{i,m} = E_{i,m} - E_{i,\text{FPA-ELM}}.$$

Positive values indicate lower error with FPA-ELM. Each point represents one of the 36 series, while the black diamond denotes the arithmetic mean across series. The distributions show a clear positive shift for MAE and RMSE. The MAPE differences are smaller for RF, matching the smaller aggregate MAPE margin.

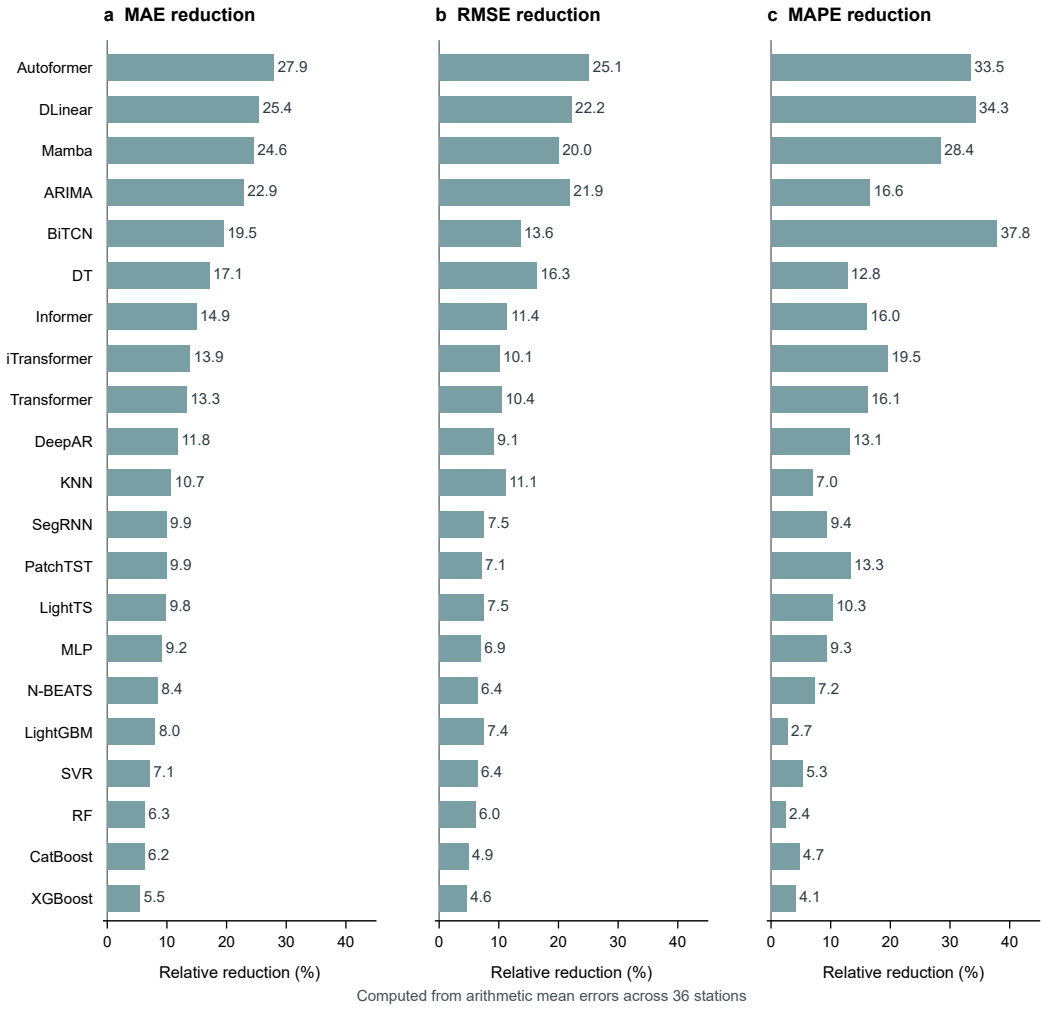


Figure 12. Relative error reductions of FPA-ELM over the 21 baseline methods.

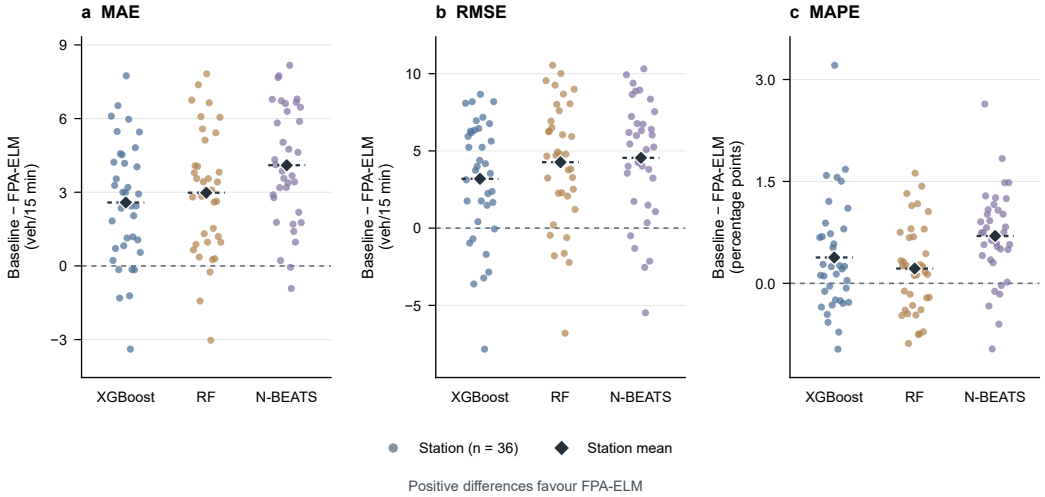


Figure 13. Series-level error differences between FPA-ELM and representative baseline methods.

Figure 14a–c compares the series-level error distributions of representative conventional and deep-learning methods with FPA-ELM. Each box contains the errors of all 36 series. The central line denotes the median, the box spans the interquartile range, and the individual points show the station-level observations. FPA-ELM shifts the distributions toward lower errors in all three metrics and shows particularly clear separation in MAE and RMSE.

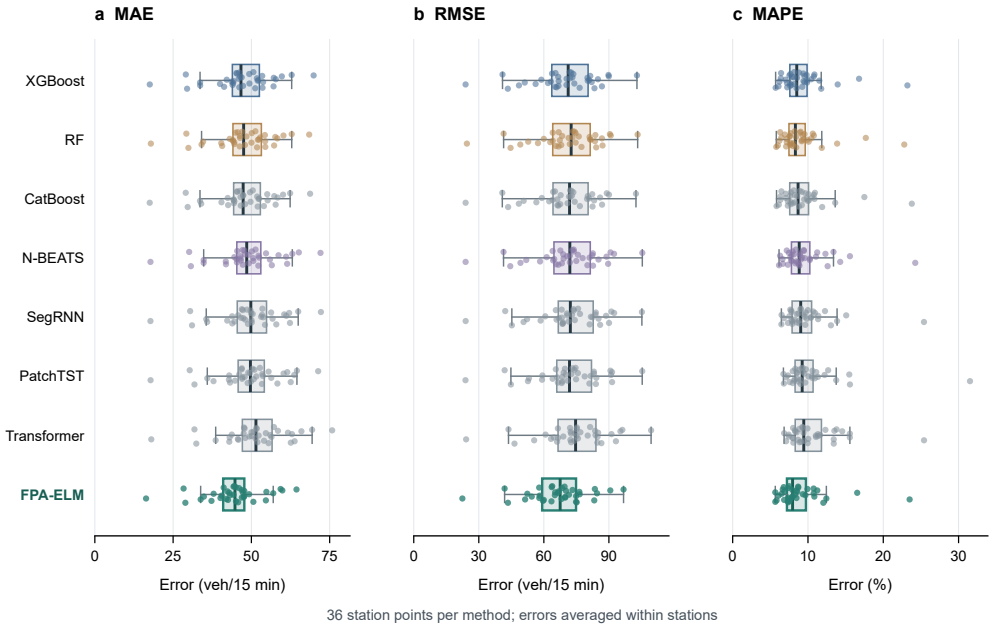


Figure 14. Series-level forecasting-error distributions of representative methods.

The performance pattern reflects the structure of one-step traffic forecasting. Short-horizon flow is dominated by strong local continuity, while the remaining variation consists of smaller departures from the current and recurring traffic levels. Tree ensembles capture nonlinear relations among recent observations effectively, which accounts for the strong performance of XGBoost, RF, and CatBoost. FPA-ELM separates these components explicitly. The fractional-prior anchor represents persistence and same-phase seasonal structure, while the ELM readout models the residual departure from this reference. The readout therefore focuses its capacity on the lower-amplitude and less regular component of the prediction problem rather than reconstructing the complete traffic-flow level.

3.6. Computational Efficiency and Complexity Analysis

We evaluated FPA-ELM against 21 representative baselines using theoretical complexity, training time, and prediction latency. All methods were run on the same platform with a single CPU thread and no GPU acceleration. Each method used the same chronological 80/20 training–test split and one-step forecasting horizon ($h = 1$). Training time covers one model fit with fixed hyperparameters and excludes hyperparameter tuning. Prediction latency is reported in microseconds per sample. Table 9 summarizes the results.

The training cost of FPA-ELM comprises fractional-prior estimation, hidden-layer feature construction, and regularized estimation of the output weights. Let n denote the number of training samples, D the input dimension, H the number of hidden neurons, and J the number of seasonal components. Fractional-prior estimation requires $\mathcal{O}(nJ^2 + J^3)$ operations, and constructing the augmented feature matrix $\mathbf{Z} = [g(\mathbf{XW} + \mathbf{b}), \mathbf{X}, \mathbf{1}]$ requires $\mathcal{O}(nDH)$ operations. With $p = H + D + 1$, the augmented QR-based ridge-regression solver requires $\mathcal{O}((n + p)p^2)$ operations. The total training complexity of one model fit is therefore

$$\begin{aligned}\mathcal{T}_{\text{train}} &= \mathcal{O}(nJ^2 + J^3 + nDH + (n+p)p^2) \\ &= \mathcal{O}\Big(nH(D+H) + H^3\Big),\end{aligned} \tag{30}$$

where the simplified form assumes fixed J and $D = \mathcal{O}(H)$. For fixed model dimensions, training cost is linear in n. FPA-ELM estimates its output weights without iterative back-propagation over multiple epochs.

At inference, fixed parameters are used to compute the fractional prior and hidden-layer representation before the linear output projection. The prediction complexity per sample is

$$\begin{aligned}\mathcal{T}_{\text{pred}} &= \mathcal{O}(J + DH + p) \\ &= \mathcal{O}(H(D+1)),\end{aligned} \tag{31}$$

which is independent of the training-set size when the model dimensions are fixed.

Table 9. Theoretical complexity, measured training time, and prediction latency at $h = 1$.

Model	Training Complexity	Prediction Complexity	Training Time (s)	Prediction (μs/Sample)
Conventional baselines				
ARIMA	$\mathcal{O}(np_A^2)$	$\mathcal{O}(p_A)$	12.91	5.79
MLP	$\mathcal{O}(EnLh_1)$	$\mathcal{O}(Lh_1 + h_1h_2)$	4.03	1.29
KNN	$\mathcal{O}(nL)$	$\mathcal{O}(nL)$	0.07	154.00
DT	$\mathcal{O}(nL\log n)$	$\mathcal{O}(\log n)$	0.64	0.20
SVR	$\mathcal{O}(n^2L)$–$\mathcal{O}(n^3)$	$\mathcal{O}(n_{\text{SV}}L)$	58.82	1106.00
RF	$\mathcal{O}(TnL\log n)$	$\mathcal{O}(T\log n)$	165.00	187.00
XGBoost	$\mathcal{O}(RnL)$	$\mathcal{O}(R\log n)$	1.59	5.85
CatBoost	$\mathcal{O}(RnL)$	$\mathcal{O}(R\log n)$	8.55	2.25
LightGBM	$\mathcal{O}(RnL)$	$\mathcal{O}(R\log n)$	4.90	123.00
Deep learning baselines				
BiTCN	$\mathcal{O}(EnP)$	$\mathcal{O}(P)$	77.57	36.34
DeepAR	$\mathcal{O}(EnP)$	$\mathcal{O}(P)$	80.17	9.61
SegRNN	$\mathcal{O}(EnP)$	$\mathcal{O}(P)$	77.05	6.30
DLinear	$\mathcal{O}(EnLC)$	$\mathcal{O}(LC)$	13.22	6.07
N-BEATS	$\mathcal{O}(EnP)$	$\mathcal{O}(P)$	16.24	0.82
Mamba	$\mathcal{O}(EnP)$	$\mathcal{O}(P)$	378.00	156.00
Transformer	$\mathcal{O}(EnL^2d)$	$\mathcal{O}(L^2d)$	77.19	71.30
PatchTST	$\mathcal{O}(EnL^2d)$	$\mathcal{O}(L^2d)$	46.74	18.97
LightTS	$\mathcal{O}(EnP)$	$\mathcal{O}(P)$	32.05	1.17
Informer	$\mathcal{O}(EnL^2d)$	$\mathcal{O}(L^2d)$	41.81	44.49
iTransformer	$\mathcal{O}(En(Ld + d^2))$	$\mathcal{O}(Ld + d^2)$	29.28	9.47
Autoformer	$\mathcal{O}(EnL^2d)$	$\mathcal{O}(L^2d)$	324.00	35.03
Proposed method				
FPA-ELM	$\mathcal{O}(nH(D+H) + H^3)$	$\mathcal{O}(H(D+1))$	3.73	6.38

Note: n is the number of training samples; L is the input sequence length; D and H are the input dimension and hidden-layer width of FPA-ELM; p_A is the autoregressive order of ARIMA; d is the embedding dimension; C is the number of channels; E is the number of training epochs; R is the number of boosting iterations; T is the number of trees; P represents the architecture-dependent computational workload per sample; n_{SV} is the number of support vectors; and h_1, h_2 are the MLP hidden-layer widths. The complexity expressions give leading-order costs and omit implementation-dependent operations. Training time excludes hyperparameter tuning. Bold identifies FPA-ELM as the proposed method.

FPA-ELM trained in 3.73 s, faster than six of the nine conventional baselines and all 12 deep learning baselines. KNN, DT, and XGBoost trained faster, whereas RF and Mamba required 165.00 s and 378.00 s, respectively. This performance reflects the non-iterative estimation of the FPA-ELM output weights.

Prediction latency was 6.38 μs per sample. FPA-ELM was faster than four conventional and eight deep learning baselines, including SVR (1106.00 μs), RF (187.00 μs), Mamba (156.00 μs), and Transformer (71.30 μs). Nine baselines achieved lower latency. FPA-ELM therefore provides single-digit-microsecond inference without ranking as the fastest method in the comparison.

4. Conclusions

This study proposed FPA-ELM for station-level traffic flow forecasting by combining a fractional-prior persistence–seasonal anchor with ELM residual correction. Fractional accumulation coefficients provide a soft prior for the seasonal coefficient profile, while the persistence coefficient and overall seasonal contribution are estimated from data. The residual learner combines fixed random nonlinear features with direct linear input connections to capture deviations from the anchor. Both stages are fitted through regularized linear systems without back-propagation.

In one-step forecasting across 36 stations, FPA-ELM achieved the lowest mean errors among the evaluated methods, with an MAE of 44.48, an RMSE of 66.42, and a MAPE of 8.92%. Relative to XGBoost, the strongest baseline in MAE and RMSE, the corresponding reductions were 5.49% and 4.59%. Ablation experiments showed that the anchor and residual learner made complementary contributions. Removing the residual learner produced the largest increase in forecasting error, while the anchor provided an informative reference for residual correction. The direct linear link further improved prediction accuracy.

The input-window analysis identified a horizon-dependent lag strategy. A longer lag block was preferred for one-step prediction, whereas compact lag inputs performed better at the longer horizons examined. Forecasting error was more sensitive to lag length than to the window used to calculate local statistics. These results support selecting lag and context windows separately, with the lag length adapted to the forecasting horizon.

FPA-ELM combines explicit modelling of persistence and seasonal structure with nonlinear residual learning in a lightweight forecasting model. The results demonstrate the predictive value of this separation for station-level traffic flow forecasting. Future work will investigate alternative seasonal priors, extend the evaluation to additional datasets, and examine training and inference costs across deployment settings.

Author Contributions: Conceptualization and methodology: J.Y. and Z.L.; formal analysis and validation: J.Y. and S.X.; writing and visualization: J.Y. and J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 62576098 and 62573122.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets supporting the findings of this study are available from the corresponding author upon reasonable request. The source code and experimental configuration files will be made publicly available upon acceptance of the manuscript to facilitate reproducibility.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Giraka, O.; Selvaraj, V.K. Short-term prediction of intersection turning volume using seasonal ARIMA model. *Transp. Lett.* **2020**, *12*, 483–490. https://doi.org/10.1080/19427867.2019.1645476.
2. Sun, S.; Zhang, C.; Yu, G. A Bayesian Network Approach to Traffic Flow Forecasting. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 124–132. https://doi.org/10.1109/TITS.2006.869623.
3. Guo, J.; Huang, W.; Williams, B.M. Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 50–64. https://doi.org/10.1016/j.trc.2014.02.006.
4. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734. https://doi.org/10.3115/v1/D14-1179.
5. Ma, X.; Tao, Z.; Wang, Y.; Yu, H.; Wang, Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 187–197. https://doi.org/10.1016/j.trc.2015.03.014.
6. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 5998–6008.
7. Zhao, C.; Ye, J.; Zhu, Z.; Huang, Y. FLRNN-FGA: Fractional-Order Lipschitz Recurrent Neural Network with Frequency-Domain Gated Attention Mechanism for Time Series Forecasting. *Fractal Fract.* **2024**, *8*, 433. https://doi.org/10.3390/fractalfract8070433.
8. Jia, Z.; Rao, N. Fractional Time-Varying Autoregressive Modeling: Parallel GAM and PINN Approaches to Dynamic Volatility Forecasting. *Fractal Fract.* **2025**, *9*, 772. https://doi.org/10.3390/fractalfract9120772.
9. Gil-Alana, L.A.; Poza, C. Seasonal Long Memory in Retail Sales in the G7 Countries. *Fractal Fract.* **2024**, *8*, 650. https://doi.org/10.3390/fractalfract8110650.
10. Chen, Y.; Wang, H.; Li, S.; Dong, R. A Novel Grey Seasonal Model for Natural Gas Production Forecasting. *Fractal Fract.* **2023**, *7*, 422. https://doi.org/10.3390/fractalfract7060422.
11. Podlubny, I. Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications. In *Mathematics in Science and Engineering*; Academic Press: San Diego, CA, USA, 1999; Volume 198.
12. Chen, T. Fractional-Order Accumulative Generation with Discrete Convolution Transformation. *Fractal Fract.* **2023**, *7*, 402. https://doi.org/10.3390/fractalfract7050402.
13. Wang, Y.; Liu, C. A New Fractional-Order Grey Prediction Model without a Parameter Estimation Process. *Fractal Fract.* **2024**, *8*, 396. https://doi.org/10.3390/fractalfract8070396.
14. Yang, B.; Zeng, X.; Zhao, J. A Novel Grey Prediction Model: A Hybrid Approach Based on Extension of the Fractional Order Discrete Grey Power Model with the Polynomial-Driven and PSO-GWO Algorithm. *Fractal Fract.* **2025**, *9*, 120. https://doi.org/10.3390/fractalfract9020120.
15. Wu, G.; Fu, H.; Jiang, P.; Chi, R.; Cai, R. Dynamic Fractional-Order Grey Prediction Model with GWO and MLP for Forecasting Overseas Talent Mobility in China. *Fractal Fract.* **2024**, *8*, 217. https://doi.org/10.3390/fractalfract8040217.
16. Ding, S.; Zhang, N.; Xu, X.; Guo, L.; Zhang, J. Deep Extreme Learning Machine and Its Application in EEG Classification. *Math. Probl. Eng.* **2015**, *2015*, 129021. https://doi.org/10.1155/2015/129021.
17. Li, Z.; Korovin, I.; Shi, X.; Gorbachev, S.; Gorbacheva, N.; Huang, W.; Cao, J. A Data-Driven Rutting Depth Short-Time Prediction Model with Metaheuristic Optimization for Asphalt Pavements Based on RIOHTrack. *IEEE/CAA J. Autom. Sin.* **2023**, *10*, 1918–1932. https://doi.org/10.1109/JAS.2023.123192.
18. Zou, W.; Xia, Y. Back propagation bidirectional extreme learning machine for traffic flow time series prediction. *Neural Comput. Appl.* **2019**, *31*, 7401–7414. https://doi.org/10.1007/s00521-018-3578-y.
19. Li, Z.; Cao, J.; Shi, X.; Huang, W. QPSO-AHES-RC: A hybrid learning model for short-term traffic flow prediction. *Soft Comput.* **2023**, *27*, 9347–9366. https://doi.org/10.1007/s00500-023-08291-w.
20. Yang, Y.; Li, Z.; Chen, J.; Liu, Z.; Cao, J. TRELM-DROP: An impavement non-iterative algorithm for traffic flow forecast. *Phys. A Stat. Mech. Its Appl.* **2024**, *633*, 129337. https://doi.org/10.1016/j.physa.2023.129337.
21. Orozco Cirilo, S.; Vargas-Canales, J.M.; Sangerman Jarquín, D.M.; Medina Cuéllar, S.E.; Antonio Bautista, J.; Valdes Cobos, A.; Rodríguez Haros, B.; Hernández Hernández, B. Fractional Long-Memory Dynamics and Residual Machine Learning for Medium-Horizon Agricultural Commodity Price Forecasting. *Fractal Fract.* **2026**, *10*, 536. https://doi.org/10.3390/fractalfract10080536.
22. Ghritlahre, H.K.; Prasad, R.K. Exergetic performance prediction of solar air heater using MLP, GRNN and RBF models of artificial neural network technique. *J. Environ. Manag.* **2018**, *223*, 566–575. https://doi.org/10.1016/j.jenvman.2018.06.033.
23. Bernaś, M.; Płaczek, B.; Porwik, P.; Pamuła, T. Segmentation of vehicle detector data for improved k-nearest neighbours-based traffic flow prediction. *IET Intell. Transp. Syst.* **2015**, *9*, 264–274. https://doi.org/10.1049/iet-its.2013.0164.
24. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*; Chapman and Hall/CRC: Boca Raton, FL, USA, 1984. https://doi.org/10.1201/9781315139470.

25. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88.
26. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. https://doi.org/10.1023/A:1010933404324.
27. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. https://doi.org/10.1145/2939672.2939785.
28. Hancock, J.T.; Khoshgoftaar, T.M. CatBoost for big data: An interdisciplinary review. *J. Big Data* **2020**, *7*, 94. https://doi.org/10.1186/s40537-020-00369-8.
29. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 3146–3154.
30. Salinas, D.; Flunkert, V.; Gasthaus, J.; Januschowski, T. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecast.* **2020**, *36*, 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001.
31. Lin, S.; Lin, W.; Wu, W.; Zhao, F.; Mo, R.; Zhang, H. SegRNN: Segment Recurrent Neural Network for Long-Term Time Series Forecasting. *IEEE Internet Things J.* **2026**, *13*, 9861–9871. https://doi.org/10.1109/JIOT.2025.3647705.
32. Zeng, A.; Chen, M.; Zhang, L.; Xu, Q. Are Transformers Effective for Time Series Forecasting? *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 11121–11128. https://doi.org/10.1609/aaai.v37i9.26317.
33. Oreshkin, B.N.; Carpov, D.; Chapados, N.; Bengio, Y. N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting. In Proceedings of the International Conference on Learning Representations, Online, 26 April–1 May 2020.
34. Gu, A.; Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In Proceedings of the First Conference on Language Modeling, Philadelphia, PA, USA, 7–9 October 2024.
35. Nie, Y.; Nguyen, N.H.; Sinthong, P.; Kalagnanam, J. A Time Series Is Worth 64 Words: Long-Term Forecasting with Transformers. In Proceedings of the International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
36. Zhang, T.; Zhang, Y.; Cao, W.; Bian, J.; Yi, X.; Zheng, S.; Li, J. Less Is More: Fast Multivariate Time Series Forecasting with Light Sampling-oriented MLP Structures. *arXiv* **2022**, arXiv:2207.01186.
37. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proc. Aaai Conf. Artif. Intell.* **2021**, *35*, 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325.
38. Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; Long, M. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
39. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In Proceedings of the Advances in Neural Information Processing Systems, Online, 6–14 December 2021; Volume 34, pp. 22419–22430.
40. Lea, C.; Flynn, M.D.; Vidal, R.; Reiter, A.; Hager, G.D. Temporal Convolutional Networks for Action Segmentation and Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1003–1012. https://doi.org/10.1109/CVPR.2017.113.

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