



Semnan University

*Journal of Modeling & Simulation in Electrical & Electronics
Engineering (MSEEE)*

Journal homepage: <https://mseee.semnan.ac.ir/>

ISSN: 2821-0786



Beyond Model Complexity: DLinear's Superior Accuracy-Efficiency Trade-off in Real-World ISP Backbone Traffic Forecasting

Hamid Reza Deihimfar¹, Ali Shahzadi¹*

Abstract-- Accurate short-term network traffic forecasting is essential for Internet Service Providers (ISPs) to support capacity planning, congestion mitigation, anomaly-aware monitoring, and timely operational decision-making. Cross-validation on both an ISP-specific dataset (CESNET-TimeSeries24) and a public benchmark (Electricity) confirms that model superiority is highly dependent on data characteristics. This study challenges the prevailing assumption that more complex deep learning architectures are inherently superior for time series forecasting. Through comprehensive experiments, we demonstrate that DLinear achieves the best accuracy-efficiency trade-off for ISP backbone traffic, while NHITS and PatchTST perform better on the Electricity dataset. These findings highlight the importance of aligning model inductive bias with domain-specific temporal patterns rather than relying solely on architectural complexity. To ensure a fair and operationally meaningful comparison, we design a unified experimental pipeline that includes consistent preprocessing, chronological data splitting, repeated runs, and evaluation across both predictive accuracy and computational efficiency. The analysis covers representative forecasting paradigms, including recurrent neural networks, decomposition-based linear models, block-based architectures, and recent Transformer-inspired approaches. The results reveal that increasing architectural complexity does not necessarily lead to superior forecasting performance in ISP traffic scenarios. In particular, DLinear emerges as the most effective and practical model on the CESNET-TimeSeries24 dataset, achieving the most favorable balance between forecasting accuracy, training cost, and inference latency. Its strong performance can be attributed to its ability to explicitly model trend and seasonal components, which aligns well with the temporal regularities commonly observed in backbone traffic. These findings challenge the assumption that more sophisticated architectures are always preferable and highlight the importance of matching model inductive bias to data characteristics and deployment

constraints. Overall, this study provides practical guidance for network operators seeking robust, efficient, and deployable forecasting solutions in resource-constrained ISP environments.

Index Terms: Network traffic forecasting, Deep learning, Transformer architectures, PatchTST, DLinear, TimesNet, iTransformer, N-BEATS, N-HITS, ISP, Time series prediction, CESNET-TimeSeries24.

I. INTRODUCTION

The rapid growth of digital services, cloud platforms, mobile applications, video streaming, Internet of Things (IoT) systems, and data-intensive enterprise services has increased the operational complexity of modern communication networks. For Internet Service Providers (ISPs), accurate network traffic forecasting is no longer a secondary optimization task, but a core requirement for proactive resource allocation, congestion avoidance, capacity planning, anomaly awareness, and Quality of Service (QoS) assurance. Early studies on Internet backbone traffic showed that long-term traffic evolution contains meaningful temporal regularities useful for forecasting and planning [31]. More recent studies have emphasized the role of machine learning in improving scalability and traffic management in next-generation networks, including 5G core infrastructures [29]. Deep learning-based traffic prediction has also been investigated as an enabling component for network optimization and intelligent control [33], while recent work in software-defined networking highlights its importance for enhancing QoS in programmable network environments [4].

Network traffic forecasting is a specialized form of time series forecasting, where future traffic volumes are predicted from historical observations. Classical statistical approaches,

Received; 2026-05-02 Revised; 2026-05-26 Accepted; 2026-06-14

1. Faculty of Electrical and Computer Engineering, Semnan University, Semnan, Iran.

*Corresponding author Email: shahzadi@semnan.ac.ir

Cite this article as:

Deihimfar, H, R and Shahzadi, A. (2026). Beyond Model Complexity: DLinear's Superior Accuracy-Efficiency Trade-off in Real-World ISP Backbone Traffic Forecasting. *Journal of Modeling & Simulation in Electrical & Electronics Engineering (MSEEE)*. Semnan University Press. 7 (2), 1-15.
DOI: <https://doi.org/10.22075/MSEEE.2026.41063.1269>

particularly autoregressive and moving-average families such as ARIMA, have long provided a foundation for forecasting and control in time series analysis [10]. Hybrid ARIMA–neural network models were later proposed to capture both linear and nonlinear structures in time series data [9]. However, real-world ISP traffic is affected by heterogeneous user behavior, application-level demand fluctuations, periodic usage patterns, abrupt changes, and possible external events. These characteristics make the problem more challenging than stationary or weakly nonlinear forecasting tasks. Consequently, machine learning strategies have become increasingly relevant because they can model complex temporal patterns beyond the assumptions of traditional statistical methods [8].

In recent years, deep learning has become a dominant paradigm for time series forecasting. Recurrent neural networks and their gated variants have been widely used to model temporal dependencies in sequential data. The gated recurrent unit introduced by Cho et al. [11] provided an efficient recurrent mechanism for sequence modeling, while subsequent deep architectures were designed to capture both long- and short-term temporal patterns [19]. Probabilistic autoregressive recurrent models, such as DeepAR, further demonstrated the value of deep recurrent architectures for uncertainty-aware forecasting across related time series [18]. Recurrent models have also been applied to groups of similar time series using clustering-based learning strategies [15]. Beyond recurrent architectures, convolutional, attention-based, graph-based, and Transformer-based models have expanded the modeling capacity of deep forecasting systems [22], [6].

Despite these advances, the relationship between architectural complexity and practical forecasting performance remains an open question. A substantial body of recent research has focused on increasingly sophisticated models, including spatio-temporal convolution networks [21], graph attention networks [24], attention-based spatial-temporal graph convolutional networks [13], multi-head spatio-temporal graph convolutional networks [30], and multi-scale spatio-temporal attention networks [32]. Other studies have explored Transformer-based forecasting models [17], [25], hybrid attention and graph neural networks [27], and spatio-temporal decoupled interactive learning [5]. Although powerful, such models often introduce considerable computational overhead, require careful tuning, and may not always provide proportional practical gains in operational ISP environments.

This issue is particularly important because forecasting models deployed in real networks must satisfy requirements beyond predictive accuracy. In ISP traffic engineering, a model should support fast training, low-latency inference, stable performance, and practical interpretability. A highly complex model that yields only marginal accuracy improvement may be less useful than a simpler model that offers competitive or superior accuracy with substantially lower computational cost. Recent comparative studies in other application domains have shown that model performance varies significantly across datasets and that no single deep architecture can be assumed to be universally superior [28]. This motivates systematic benchmarking under realistic traffic forecasting conditions.

Another relevant direction concerns data characteristics and model inductive bias. Decomposition-based forecasting methods explicitly separate time series into components such

as trend, seasonality, and residual patterns, potentially simplifying the learning task [26]. Similar decomposition and entropy-based strategies have been used in other domains, such as wind speed forecasting [34]. Hybrid deep learning frameworks have also been developed to combine complementary modeling strengths [20], [35]. In addition, recent studies have considered external factors and inner–outer correlations in time series forecasting [14], while data augmentation methods have been surveyed as a way to improve deep learning performance when training data are limited or imbalanced [16]. These studies suggest that predictive success depends not only on model depth or architectural novelty but also on the alignment between model assumptions, data structure, and operational constraints.

Motivated by these observations, this study provides a comprehensive empirical evaluation of eight representative deep learning models for real-world ISP network traffic prediction using the CESNET-TimeSeries24 dataset. The main objective is to examine whether complex architectures necessarily outperform simpler models when both forecasting accuracy and computational efficiency are considered. Unlike studies focused primarily on prediction error, this work evaluates models from a multi-dimensional perspective, including accuracy metrics, training time, inference latency, and the trade-off between predictive performance and computational cost.

The main contributions of this paper are summarized as follows:

- We provide an empirical benchmark of eight deep learning models for real-world ISP traffic forecasting using the CESNET-TimeSeries24 dataset.
- We evaluate not only forecasting accuracy but also computational efficiency, including training time and inference latency, which are essential for practical deployment.
- We analyze the trade-off between model complexity, predictive performance, and operational cost in the context of ISP traffic prediction.
- We provide practical insights for network operators and researchers regarding the selection of forecasting models under real-world resource and latency constraints.

Recent advances in deep time series forecasting have predominantly focused on increasing architectural complexity through attention mechanisms, hierarchical structures, and multi-scale representations. However, this study offers a different perspective by demonstrating that superior performance can be achieved by carefully aligning model inductive bias with the inherent characteristics of the target domain. Specifically, on ISP backbone traffic, DLinear’s explicit decomposition into trend and seasonal components provides a strong inductive bias that closely matches the pronounced daily and weekly periodicities driven by human activity patterns. This alignment enables DLinear to outperform significantly more complex architectures while maintaining dramatically lower computational requirements. By conducting experiments on both ISP-specific and general-purpose datasets, the present work contributes not only a practical benchmarking study but also a conceptual insight: in operational network environments, domain-aware

inductive biases often deliver better accuracy-efficiency trade-offs than generic increases in model sophistication.

The remainder of this paper is organized as follows. Section I reviews related work on time series forecasting, network traffic prediction, attention-based models, graph neural networks, hybrid forecasting, and comparative evaluation. Section II describes the dataset, preprocessing steps, forecasting models, experimental setup, and evaluation metrics. Section III presents the experimental results. Section IV discusses the findings, practical implications, limitations, and future research directions. Finally, Section V concludes the paper.

II. RELATED WORK

This section reviews the main research directions relevant to the present study. The discussion is organized into six parts: traditional and machine learning-based time series forecasting; deep learning for temporal modeling; network traffic prediction; spatio-temporal and graph-based forecasting; attention and Transformer-based models; and hybrid, decomposition-based, and comparative forecasting approaches.

A. Time Series Forecasting: From Statistical Models to Machine Learning

Time series forecasting has traditionally relied on statistical models that describe temporal dependence through autoregressive, moving-average, and integrated components. ARIMA and related methods remain widely used because of their interpretability and effectiveness in modeling linear temporal structures [10]. However, many real-world series contain nonlinear dependencies, multiple seasonalities, abrupt transitions, and interactions that are difficult to capture using purely linear assumptions. To address this limitation, Zhang [9] proposed a hybrid ARIMA-neural network model, showing that statistical and neural components can jointly capture linear and nonlinear patterns.

The emergence of machine learning introduced a broader set of forecasting strategies. Bontempi et al. [8] reviewed machine learning strategies for time series forecasting and emphasized supervised learning formulations, feature construction, and model selection. Compared with classical statistical methods, machine learning models can better approximate nonlinear relationships. However, they often require careful feature engineering and may struggle with long-range temporal dependencies when temporal structure is not explicitly modeled. These limitations motivated the transition toward deep learning architectures capable of representation learning directly from sequential data.

B. Deep Learning for Time Series Forecasting

Deep learning models have substantially changed time series forecasting by enabling automatic extraction of hierarchical temporal representations. Recurrent neural networks are among the earliest deep learning architectures designed for sequential data. The encoder-decoder framework and gated recurrent mechanisms introduced by Cho et al. [11] provided a foundation for efficient recurrent sequence modeling. Subsequent studies developed deep neural architectures capable of modeling both long-term and short-term temporal dependencies [19], which is particularly important in traffic

forecasting, where daily, weekly, and bursty patterns may coexist.

Probabilistic deep forecasting has also received considerable attention. DeepAR, proposed by Salinas et al. [18], uses autoregressive recurrent networks to generate probabilistic forecasts and is useful when uncertainty estimates are required. Bandara et al. [15] investigated recurrent neural networks across groups of similar time series and demonstrated the value of exploiting shared patterns among related series. Broader reviews have reported strong performance of recurrent, convolutional, attention-based, and hybrid architectures across multiple domains [22]. In financial forecasting, Sezer et al. [6] also highlighted the growing dominance of deep learning methods, while noting that model performance depends strongly on data characteristics and experimental design.

C. Network Traffic Prediction

Network traffic forecasting is a domain-specific time series forecasting problem with direct implications for network management. Early work on Internet backbone traffic forecasting showed that traffic traces exhibit long-term patterns that can be exploited for prediction and capacity planning [31]. With the development of mobile and 5G networks, traffic forecasting has become increasingly important for scalability and resource orchestration. Alawe et al. [29] studied machine learning-based forecasting for 5G core network scalability, demonstrating the potential of predictive models in next-generation network management.

Deep learning-based network traffic prediction has also been investigated for optimization and control. Troia et al. [33] explored deep learning techniques for traffic prediction in network optimization scenarios, showing that learned temporal representations can support proactive decision-making. More recently, Mahdi and Al Saadi [4] studied deep learning-based smart traffic prediction for enhancing QoS in software-defined networking. These studies indicate that accurate forecasting can improve network planning, congestion management, and service reliability. However, many existing works focus mainly on accuracy and provide limited analysis of computational cost, inference latency, and deployment feasibility.

D. Spatio-Temporal and Graph-Based Traffic Forecasting

Many traffic forecasting problems involve both temporal dynamics and spatial or topological dependencies. Although road traffic and ISP network traffic differ in physical interpretation, both may exhibit structured dependencies among locations, nodes, or links. Zhang et al. [12] proposed deep spatio-temporal residual networks for citywide crowd flow prediction, demonstrating the effectiveness of deep architectures in jointly modeling spatial and temporal dependencies. Li et al. [21] introduced an adaptive spatial-temporal convolution network for traffic forecasting, further emphasizing dynamic spatial-temporal feature extraction.

Graph neural networks and graph attention mechanisms have become prominent tools for modeling structured dependencies. Guo et al. [13] developed attention-based spatial-temporal graph convolutional networks for traffic flow forecasting, combining graph convolution with attention to capture complex spatial-temporal relationships. Zhao et al.

[24] proposed spatio-temporal graph attention networks for urban traffic flow prediction, while Wu et al. [23] introduced a graph attention LSTM model that combines graph-based spatial reasoning with recurrent temporal modeling. More recent models extend this direction through multi-head attention and graph convolution [30], as well as multi-scale spatio-temporal attention networks for network-scale traffic learning and forecasting [32]. Chen and Wu [5] further investigated spatio-temporal decoupled interactive learning, reflecting the continuing trend toward specialized and complex architectures.

Although these models can be effective when spatial topology is central to the forecasting task, their complexity may create challenges for real-time ISP deployment. In operational settings, the objective is not only to capture possible dependencies but also to produce sufficiently accurate forecasts under strict computational and latency constraints. Therefore, complex spatio-temporal models should be compared with simpler alternatives under realistic conditions.

E. Attention-Based and Transformer-Based Forecasting Models

Attention mechanisms have been widely adopted in time series forecasting because they allow models to focus on the most relevant historical observations. Shih et al. [7] proposed temporal pattern attention for multivariate time series forecasting, showing that attention can improve the modeling of variable-specific temporal dependencies. Attention has also become central in graph-based and spatio-temporal forecasting models, where it helps identify important nodes, time steps, or feature interactions [13], [30], [32].

Transformer-based models have further expanded the use of attention in time series forecasting. The Temporal Fusion Transformer proposed by Lim et al. [17] introduced an interpretable architecture for multi-horizon forecasting, combining attention mechanisms with variable selection and gating structures. Okamoto [25] provides a comprehensive survey of Transformer-based time series forecasting models, showing that these architectures have become a major research direction. Cheng et al. [27] also investigated hybrid attention mechanisms with graph neural networks, reflecting growing interest in combining attention, graph learning, and temporal modeling.

Despite their strong representational capacity, attention-based and Transformer-based models often have higher computational requirements than simpler models. Their performance can also depend heavily on dataset size, forecasting horizon, sequence length, and tuning strategy. Thus, their practical value in ISP traffic prediction should be assessed jointly in terms of accuracy and efficiency.

F. Hybrid Models, Decomposition, External Factors, and Data Augmentation

Hybrid forecasting models aim to combine complementary strengths of different approaches. The hybrid ARIMA–neural network model proposed by Zhang [9] is an early example of this strategy. More recent hybrid deep learning models have continued this direction by integrating multiple neural components or combining time series analysis with deep learning [20], [35]. These models are motivated by the fact that different architectures may capture different aspects of

the data, including linear trends, nonlinear residuals, seasonality, and abrupt changes.

Decomposition-based methods constitute another important line of research. By decomposing a time series into trend, seasonal, and residual components, the forecasting task may become easier and more interpretable. Ge et al. [26] revisited decomposition for time series forecasting by focusing on temporal patterns, while Mezaache et al. [34] integrated decomposition and deep learning models for wind speed forecasting. Although these studies are not limited to network traffic, they are relevant because ISP traffic often contains periodic and trend-like structures that may benefit from explicit or implicit decomposition.

Recent work has also explored integrating external factors and correlation structures. Nguyen Nguyen et al. [14] proposed a framework for incorporating external factors with inner and outer correlations in time series forecasting. Such approaches are important because network traffic may be influenced by external events, service usage patterns, and interactions among traffic sources. In parallel, data augmentation has been studied to improve deep learning robustness in time series tasks. Wen et al. [16] surveyed augmentation methods for time series deep learning and emphasized their value when data availability or diversity is limited.

G. Comparative Evaluation and Research Gap

The rapid growth of forecasting architectures has made comparative evaluation increasingly important. Maragos and Refanidis [28] provided a comparative evaluation of time-series forecasting models for energy datasets, showing that model ranking can vary across domains and datasets. This observation is highly relevant to network traffic prediction: a model that performs well on road traffic, energy consumption, financial data, or synthetic benchmarks may not necessarily be optimal for ISP traffic. Therefore, domain-specific benchmarking is required.

Existing studies have advanced statistical forecasting, machine learning-based forecasting, deep recurrent models, graph neural networks, attention mechanisms, Transformer architectures, hybrid models, decomposition-based approaches, and data augmentation. However, three limitations remain evident. First, many studies emphasize predictive accuracy while paying limited attention to training cost, inference latency, and operational efficiency. Second, much of the spatio-temporal traffic forecasting literature focuses on urban traffic, crowd flow, or road-network data rather than ISP traffic traces. Third, increasingly complex architectures are often compared without a systematic analysis of whether their additional complexity is justified in real-world deployment scenarios.

To address these gaps, the present study conducts a comprehensive empirical evaluation of eight representative deep learning models on a real-world ISP traffic forecasting dataset. Unlike studies that focus only on error reduction, this work jointly considers forecasting accuracy, training time, inference latency, and the practical trade-off between model complexity and operational utility. Accordingly, it aims to provide evidence-based guidance for selecting forecasting models suitable for real-world ISP traffic prediction.

III. METHODOLOGY

This section presents the methodological framework for evaluating and comparing representative deep learning models for ISP network traffic forecasting. The protocol is designed to ensure fairness, reproducibility, and practical relevance across architectures with different modeling principles. It includes the research design, dataset description, preprocessing pipeline, model selection, hyperparameter control, experimental configuration, and evaluation criteria. The core setup, including forecasting formulation, preprocessing flow, and evaluation protocol, follows the source manuscript [25].

A. Research Design

The primary objective of this study is to conduct a multidimensional comparative analysis of deep learning architectures for short-term network traffic forecasting. In addition to predictive accuracy, the proposed evaluation framework explicitly considers computational efficiency and deployment feasibility, both of which are essential for operational Internet Service Provider (ISP) environments.

The experimental pipeline is organized into six main stages:

1. **Model selection:** A set of representative forecasting models is selected from decomposition-based, recurrent, and Transformer-inspired model families.
2. **Dataset selection:** A real-world ISP traffic dataset is employed to capture practical characteristics of network traffic, including periodicity, burstiness, temporal dependence, and non-stationarity.
3. **Standardized preprocessing:** All candidate models are trained and evaluated using identically preprocessed data to ensure a fair and reproducible comparison.
4. **Unified experimental protocol:** Core experimental conditions, including the forecasting horizon, optimizer configuration, training procedure, and early-stopping criterion, are kept consistent across models to the greatest extent possible.
5. **Accuracy–efficiency evaluation:** Each model is assessed using both predictive performance metrics and computational efficiency indicators.
6. **Repeated execution:** Each model is executed under multiple random seeds to reduce the effect of stochastic variation and improve the reliability of the reported results.

Formally, let $\mathcal{M} = \{M_1, M_2, \dots, M_K\}$ denote the set of candidate forecasting models, and let $D_{\text{tr}}, D_{\text{va}}, D_{\text{te}}$ represent the chronological training, validation, and test sets, respectively. For each model $M_k \in \mathcal{M}$, the trainable parameters θ_k are optimized by minimizing the training loss over the training set. Given a hyperparameter configuration λ_k , this optimization can be written as

$$\theta_k^*(\lambda_k) = \underset{\theta_k}{\operatorname{argmin}} \mathcal{L}_{\text{tr}}(M_k(D_{\text{tr}}; \theta_k, \lambda_k), D_{\text{tr}}), \quad (1)$$

where $\theta_k^*(\lambda_k)$ denotes the optimized model parameters obtained under the hyperparameter setting λ_k , and $\mathcal{L}_{\text{tr}}$ is the training loss function.

The optimal hyperparameter configuration for each model is then selected according to its validation performance:

$$\lambda_k^\dagger = \underset{\lambda_k \in \Lambda_k}{\operatorname{argmin}} \mathcal{J}_{\text{val}}(M_k(D_{\text{va}}; \theta_k^*(\lambda_k), \lambda_k), D_{\text{va}}), \quad (2)$$

where Λ_k denotes the hyperparameter search space of the model M_k , and $\mathcal{J}_{\text{val}}$ is the validation objective function. Accordingly, the final selected parameters of the model M_k are defined as $\theta_k^\dagger = \theta_k^*(\lambda_k^\dagger)$.

After training and validation, the final comparison is performed on the held-out test set D_{te} . The test-stage evaluation considers both forecasting accuracy and computational efficiency, enabling a comprehensive assessment of the trade-offs among the candidate models. This design ensures that the comparison reflects not only differences in predictive performance but also the practical suitability of each architecture for real-world ISP traffic forecasting.

B. Dataset Description

The experiments are conducted on the **CESNET-TimeSeries24** dataset, a real-world traffic dataset derived from ISP backbone measurements. The selected series represent aggregated traffic volume in bits per second (bps), sampled at regular 5-minute intervals over a 24-month period [25]. Such a dataset is suitable for benchmarking because it exhibits realistic traffic dynamics, including daily and weekly seasonality, trend components, and bursty fluctuations.

To preserve temporal causality and avoid future information leakage, the dataset is split chronologically into training, validation, and test subsets:

- **Training set:** 70%
- **Validation set:** 15%
- **Testing set:** 15%

Let $\{x_t\}_{t=1}^T$ denote the normalized univariate traffic sequence. The forecasting task is formulated as a supervised multi-step prediction problem. Given a historical window of length L , the goal is to predict the next H traffic values:

$$\hat{\mathbf{y}}_{t+1:t+H} = f_\theta(\mathbf{x}_{t-L+1:t}), \quad (3)$$

where

$$\mathbf{x}_{t-L+1:t} = [x_{t-L+1}, x_{t-L+2}, \dots, x_t]^\top \in \mathbb{R}^L, \quad (4)$$

and

$$\hat{\mathbf{y}}_{t+1:t+H} = [\hat{x}_{t+1}, \hat{x}_{t+2}, \dots, \hat{x}_{t+H}]^\top \in \mathbb{R}^H. \quad (6)$$

In this study, the input length is fixed at $L = 96$, corresponding to 8 hours of historical observations, and the prediction horizon is set to $H = 12$, corresponding to 1 hour ahead forecasting under 5-minute sampling [25]. Therefore, the sample set can be written as

$$\mathcal{S} = \{(\mathbf{x}_{t-L+1:t}, \mathbf{y}_{t+1:t+H})\}_{t=L}^{T-H}. \quad (6)$$

This formulation provides sufficient temporal context while maintaining an operationally meaningful forecasting lead time for network management tasks.

C. Data Preprocessing

A consistent preprocessing pipeline is applied to all models before training and evaluation. This ensures comparability across architectures and improves optimization stability.

1) Missing Value Imputation

Real-world traffic measurements may contain occasional missing observations due to telemetry interruptions, collection failures, or storage inconsistencies. Missing values are handled using linear interpolation [25]. If a value at time t is missing and bounded by observed values at times t_a and t_b , with $t_a < t < t_b$, then the imputed value is computed as

$$x_t = x_{t_a} + \frac{t - t_a}{t_b - t_a}(x_{t_b} - x_{t_a}). \quad (7)$$

This approach is computationally simple and preserves local continuity in the traffic signal without introducing abrupt artifacts.

2) Normalization

To stabilize optimization and reduce scale sensitivity, the traffic series is normalized to $[0,1]$ using Min–Max normalization:

$$x'_t = \frac{x_t - x_{\min}}{x_{\max} - x_{\min}}, \quad (8)$$

where $x_{\min}$ and $x_{\max}$ are estimated exclusively from the training set [25]. This is essential to prevent information leakage from validation and test periods. The inverse transformation used for restoring the original scale is

$$x_t = x'_t(x_{\max} - x_{\min}) + x_{\min}. \quad (9)$$

3) Sequence Generation

After normalization, the univariate traffic series is transformed into overlapping input–output pairs using a sliding-window operator. For each valid time index t , the generated sample is

$$\Phi_t = (x_{t-L+1:t}, y_{t+1:t+H}). \quad (10)$$

The number of generated samples is therefore

$$N = T - L - H + 1. \quad (11)$$

This procedure yields a unified supervised learning setup for all forecasting models and ensures that each architecture is trained on the same information content.

D. Deep Learning Models

This study evaluates eight forecasting architectures selected to represent major methodological families in modern time-series prediction. The set includes decomposition-based linear models, recurrent sequence models, and recent architectures based on attention or multi-scale temporal modeling. The descriptions below follow the original methodological structure [25], with a concise technical characterization of each model class.

1) DLinear

DLinear is a decomposition-based forecasting model that separates the input sequence into trend and seasonal components and models each part with linear mappings. Let $\mathbf{x} \in \mathbb{R}^L$ denote the input window. The decomposition can be abstracted as

$$\mathbf{x} = \mathbf{x}^{(\text{trend})} + \mathbf{x}^{(\text{season})}, \quad (12)$$

and the forecast is obtained by

$$\hat{\mathbf{y}} = W_t \mathbf{x}^{(\text{trend})} + W_s \mathbf{x}^{(\text{season})} + \mathbf{b}, \quad (13)$$

where W_t and W_s are learnable linear projection matrices. Its main advantage is low computational complexity, making it an efficiency-oriented reference model.

2) PatchTST

PatchTST is a Transformer-based forecasting model that segments the input sequence into subseries patches before applying self-attention. Given patch length P and stride S , the number of patches is approximately

$$N_p = \left\lfloor \frac{L - P}{S} \right\rfloor + 1. \quad (14)$$

Each patch is embedded into a latent representation and processed through multi-head self-attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V. \quad (15)$$

This design reduces effective token length while preserving local and global temporal dependencies, which is desirable for ISP traffic with repetitive local structures and broader periodic behavior.

3) iTransformer

iTransformer reformulates Transformer-based forecasting by altering the conventional tokenization perspective and emphasizing dependency modeling in a transformed representation space. In general form, its encoder produces

$$\mathbf{Z} = \text{TransformerEncoder}(E(\mathbf{x})), \quad (16)$$

where $E(\cdot)$ denotes the embedding operator, and $\mathbf{Z}$ is the latent sequence representation. The final forecast is generated through a projection head,

$$\hat{\mathbf{y}} = W_o \mathbf{Z} + \mathbf{b}_o. \quad (17)$$

It is included because it represents a recent architectural development with competitive forecasting performance.

4) TimesNet

TimesNet is designed to capture multi-periodic temporal variations by transforming one-dimensional sequences into two-dimensional temporal representations and applying convolutional or Inception-style operations. Let $\mathbf{x} \in \mathbb{R}^L$ be

the input sequence. A periodic reshaping operation maps it into $\mathcal{X} \in \mathbb{R}^{P \times Q}$, $PQ \approx L$, where different values of P and Q correspond to candidate periodic structures. Temporal features are then extracted through multi-scale convolutional blocks:

$$\mathbf{h}^{(\ell+1)} = \sigma(\text{Conv}^{(\ell)}(\mathbf{h}^{(\ell)})). \quad (18)$$

This architecture is particularly relevant for traffic data with recurring patterns at multiple temporal scales.

5) N-BEATS and N-HiTS

N-BEATS is a deep forecasting architecture based on stacked fully connected blocks that iteratively perform backcasting and forecasting. For block b , the computation can be summarized as

$$\mathbf{r}^{(b+1)} = \mathbf{r}^{(b)} - \mathbf{b}^{(b)} \quad (19)$$

$$\hat{\mathbf{y}}^{(b+1)} = \hat{\mathbf{y}}^{(b)} + \mathbf{f}^{(b)} \quad (20)$$

where $\mathbf{r}^{(b)}$ is the residual input, $\mathbf{b}^{(b)}$ is the backcast component, and $\mathbf{f}^{(b)}$ is the forecast contribution. N-HiTS extends this principle through hierarchical interpolation and multi-rate sampling, improving representation efficiency for multi-step forecasting. These models provide strong neural forecasting capability without relying on recurrence or attention.

6) LSTM and GRU

LSTM and GRU are established recurrent architectures for sequential modeling. Their gating mechanisms are designed to mitigate vanishing gradients and preserve temporal memory. For LSTM, the state transitions are defined as

$$\begin{aligned} \mathbf{i}_t &= \sigma(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + \mathbf{b}_i), \\ \mathbf{f}_t &= \sigma(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + \mathbf{b}_f), \\ \mathbf{o}_t &= \sigma(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + \mathbf{b}_o), \\ \tilde{\mathbf{c}}_t &= \tanh(W_c \mathbf{x}_t + U_c \mathbf{h}_{t-1} + \mathbf{b}_c), \\ \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \\ \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t). \end{aligned} \quad (21)$$

For GRU, the recurrence is

$$\begin{aligned} \mathbf{z}_t &= \sigma(W_z \mathbf{x}_t + U_z \mathbf{h}_{t-1} + \mathbf{b}_z), \\ \mathbf{r}_t &= \sigma(W_r \mathbf{x}_t + U_r \mathbf{h}_{t-1} + \mathbf{b}_r), \\ \tilde{\mathbf{h}}_t &= \tanh(W_h \mathbf{x}_t + U_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h), \\ \mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t. \end{aligned} \quad (22)$$

Because of their historical importance and continued practical relevance, they serve as strong baselines for deep learning in this study.

E. Hyperparameter Optimization and Model Capacity Control

To ensure a fair and reproducible comparison, all models were trained under the same general protocol, including identical input length, forecasting horizon, optimizer, batch size, maximum number of epochs, early stopping criterion, and loss function. Model-specific hyperparameters were

selected using validation-based tuning, with a limited search budget kept comparable across model families. The final configuration for each model was chosen according to the lowest validation Mean Squared Error (MSE):

$$\lambda_k^* = \underset{\lambda \in \Lambda_k}{\operatorname{argmin}} \text{MSE}_{\text{val}}(M_k; \lambda), \quad (23)$$

where Λ_k denotes the search space of the model M_k .

For recurrent and convolutional baselines, the tuning space mainly included the number of layers, hidden dimension, dropout rate, and kernel-related settings. For Transformer-based models, including PatchTST, TimesNet, and iTransformer, the search covered embedding dimension, number of attention heads, encoder depth, dropout, and model-specific parameters such as patch length. For N-BEATS and N-HiTS, the number of blocks, the layers per block, the hidden dimension, and dropout were tuned.

To avoid unfair advantages induced by excessive model capacity, configurations were selected to balance predictive performance and computational cost. In addition, the approximate number of trainable parameters for each model was recorded and is reported in Table I. If θ_k denotes the parameter vector of model M_k , then its capacity is quantified by $\Pi_k = |\theta_k|$.

This enables interpretation not only in terms of forecasting accuracy, but also with respect to model complexity and efficiency.

TABLE I.
Hyperparameter Search Space for the Evaluated Forecasting mModel Families

Model family	Search space
Recurrent models	Layers $\in \{1,2,3\}$; hidden dimension $\in \{64,128,256\}$; dropout $\in \{0.1,0.2,0.3\}$
Convolutional models	Number of layers $\in \{2,3,4\}$; hidden channels $\in \{64,128,256\}$; kernel size $\in \{3,5,7\}$; dropout $\in \{0.1,0.2,0.3\}$
Transformer-based models	Encoder layers $\in \{2,3,4\}$; embedding dimension $\in \{128,256,512\}$; heads $\in \{4,8\}$; dropout $\in \{0.1,0.2,0.3\}$
PatchTST	Patch length $\in \{8,16,24\}$; stride $\in \{4,8,12\}$; embedding dimension $\in \{128,256,512\}$; heads $\in \{4,8\}$
N-BEATS / N-HiTS	Blocks $\in \{2,3,4\}$; layers per block $\in \{2,3,4\}$; hidden dimension $\in \{128,256,512\}$; dropout $\in \{0.0,0.1,0.2\}$

To ensure a fair and reproducible comparison, all models were trained under the same general protocol. Model-specific hyperparameters were optimized using **random search with 30 independent trials per model**. This approach balances exploration and computational cost while maintaining comparable tuning effort across architectures. The final configuration for each model was selected based on the lowest validation MSE.

The detailed hyperparameter search spaces are provided in Table I. The best hyperparameters found for each model are summarized in the updated Table I (bottom part). To avoid unfair advantages from excessive model capacity, we constrained the number of trainable parameters to a

reasonable range (see Table II). This transparent, standardized tuning protocol enhances the reproducibility and fairness of the benchmark.

F. Experimental Setup

All experiments are implemented in **PyTorch** and executed on a single **NVIDIA A100 GPU** [25]. To ensure fairness, all models are trained under a unified set of core hyperparameters whenever applicable.

The shared experimental settings are as follows:

- **Input sequence length:** $L = 96$
- **Forecast horizon:** $H = 12$
- **Optimizer:** Adam
- **Learning rate:** 10^{-4}
- **Batch size:** 32
- **Maximum number of epochs:** 50
- **Early stopping:** enabled based on validation performance
- **Loss function:** Mean Squared Error (MSE)

The training objective is

$$\mathcal{L}_{\text{MSE}} = \frac{1}{NH} \sum_{n=1}^N \sum_{h=1}^H (y_{n,h} - \hat{y}_{n,h})^2, \quad (24)$$

where $y_{n,h}$ and $\hat{y}_{n,h}$ denote the true and predicted values of sample n at forecasting step h . Parameter updates follow the Adam recursion:

$$\begin{aligned} \mathbf{m}_t &= \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \mathbf{g}_t, \\ \mathbf{v}_t &= \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) \mathbf{g}_t^2, \\ \theta_t &= \theta_{t-1} - \eta \frac{\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon}, \end{aligned} \quad (25)$$

where $\mathbf{g}_t = \nabla_{\theta} \mathcal{L}_{\text{MSE}}$ is the stochastic gradient and η is the learning rate.

Although the global training protocol is standardized, architecture-specific hyperparameters are tuned through validation-based tuning [25]. To improve robustness, each experiment is repeated five times using different random seeds, and the reported values are aggregated across runs. Let $s = 1, \dots, R$ denote the run index with $R = 5$. The mean performance of a metric Q is $\bar{Q} = \frac{1}{R} \sum_{s=1}^R Q^{(s)}$, and its standard deviation is $\sigma_Q = \sqrt{\frac{1}{R-1} \sum_{s=1}^R (Q^{(s)} - \bar{Q})^2}$.

G. Evaluation Metrics

Model performance is evaluated from two complementary perspectives: **forecasting accuracy** and **computational efficiency**. This dual evaluation is essential because forecasting models in ISP environments must be both accurate and operationally efficient.

1) Accuracy Metrics

The following regression metrics are used to quantify predictive performance:

a) Root Mean Squared Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{NH} \sum_{n=1}^N \sum_{h=1}^H (y_{n,h} - \hat{y}_{n,h})^2}. \quad (26)$$

b) Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{NH} \sum_{n=1}^N \sum_{h=1}^H |y_{n,h} - \hat{y}_{n,h}|. \quad (27)$$

c) R^2 Score

$$R^2 = 1 - \frac{\sum_{n=1}^N \sum_{h=1}^H (y_{n,h} - \hat{y}_{n,h})^2}{\sum_{n=1}^N \sum_{h=1}^H (y_{n,h} - \bar{y})^2}, \quad (28)$$

$$\text{Where } \bar{y} = \frac{1}{NH} \sum_{n=1}^N \sum_{h=1}^H y_{n,h}.$$

For completeness, the horizon-wise error at step h can also be written as

$$\text{RMSE}(h) = \sqrt{\frac{1}{N} \sum_{n=1}^N (y_{n,h} - \hat{y}_{n,h})^2}, \quad (29)$$

which is useful when analyzing error propagation over the forecast horizon.

2) Efficiency Metrics

To complement predictive accuracy, the following efficiency-oriented metrics are measured:

- **Training Time:** total optimization time in seconds
- **Inference Latency:** average prediction time in milliseconds

If T_{train} denotes the total training time and N_{inf} denotes the number of evaluated inference calls, the average latency is computed as $\text{Latency} = \frac{T_{\text{inf}}}{N_{\text{inf}}}$.

These measures help determine whether highly accurate models remain practical for real-time or near-real-time deployment.

H. Accuracy-Efficiency Trade-off Analysis

Beyond reporting raw metrics, this study explicitly analyzes the trade-off between forecasting accuracy and computational efficiency. In practice, the most accurate model is not always the most appropriate for deployment. ISP operators often require a balance between prediction quality, retraining cost, hardware consumption, and response latency. Therefore, models are interpreted not only by their ranking on RMSE, MAE, and R^2 , but also by the computational burden associated with training and inference.

This trade-off can be expressed conceptually through a joint utility function

$$\mathcal{U}(M_k) = \alpha \tilde{\mathcal{A}}(M_k) - \beta \tilde{\mathcal{C}}(M_k), \quad (30)$$

where $\tilde{\mathcal{A}}$ denotes a normalized accuracy score, $\tilde{\mathcal{C}}$ denotes a normalized computational cost, and $\alpha, \beta \geq 0$ reflect application priorities. Although the present study reports the

underlying metrics separately rather than collapsing them into a single score, this formulation clarifies the decision-making perspective behind the analysis.

To provide a more structured evaluation of the accuracy-efficiency trade-off, we define a weighted utility score as: $\text{Utility} = 0.7 \times (1 - \text{normalized RMSE}) - 0.3 \times \text{normalized Training Time}$. This score prioritizes predictive accuracy while penalizing high computational cost, reflecting practical ISP requirements. Fig. 4 visualizes the Pareto front, clearly showing that DLinear dominates the trade-off space. The justification for this analysis lies in the operational reality of ISP environments, where both accuracy and resource efficiency are critical for sustainable deployment.

I. Statistical and Practical Interpretation

To support reliable conclusions, the reported results include variability across repeated runs. In particular, standard deviations are considered alongside mean performance values in order to assess stability and consistency [25]. This is important because two models with similar average accuracy may differ substantially in robustness.

From a practical ISP management perspective, a model with slightly lower forecasting accuracy but substantially lower latency or training cost may be preferable when fast adaptation and scalable deployment are required. Consequently, the final analysis emphasizes both statistical validity and operational usefulness.

IV. RESULTS

This section presents the empirical evaluation of eight forecasting models, namely DLinear, GRU, LSTM, PatchTST, iTransformer, N-BEATS, N-HiTS, and TimesNet, on the CESNET-TimeSeries24 dataset. The comparison focuses on two main dimensions: predictive accuracy and computational efficiency. Predictive performance is assessed using RMSE, MAE, and R^2 , while computational performance is evaluated in terms of training time and inference latency. The objective is to identify the model that provides the most suitable balance between forecasting quality and operational feasibility for ISP traffic forecasting.

Overall, DLinear achieved the strongest performance across all evaluated criteria. Its decomposition-based linear architecture proved highly effective for ISP traffic data, which typically exhibits regular temporal dynamics, trend behavior, and recurring seasonal patterns. This result suggests that the success of a forecasting model depends not only on its representational capacity, but also on the degree to which its inductive bias matches the statistical structure of the data.

A. Overall Model Performance

Table II reports the mean and standard deviation of all evaluation metrics over five independent runs. Among all compared methods, DLinear achieved the best overall results, with the lowest RMSE (0.7231 ± 0.012), the lowest MAE (0.2543 ± 0.008), and the highest R^2 (0.9838 ± 0.0004). These values indicate that DLinear provided the most accurate forecasts while maintaining high stability across repeated runs.

GRU and LSTM ranked second and third, respectively, indicating that recurrent architectures remain competitive for traffic forecasting tasks. GRU achieved an R^2 of 0.9771 and LSTM achieved 0.9756, both showing strong predictive capability, but still remaining below DLinear. In contrast, the more complex architectures, including PatchTST, iTransformer, N-BEATS, N-HiTS, and TimesNet, did not surpass DLinear in any of the considered accuracy metrics. TimesNet showed the weakest predictive performance, obtaining the highest error values and the lowest R^2 score of 0.9566. Fig. 1 provides a visual overview of the comparative results and confirms the leading position of DLinear.

TABLE II.

Performance Comparison Across all Evaluated Models (Mean $\pm$ Standard Deviation over 5 Runs)

Model	RMSE $\downarrow$	MAE $\downarrow$	$R^2 \uparrow$	Training Time (s) $\downarrow$	Inference (ms) $\downarrow$	Peak GPU Memory (GB)
DLinear	0.7231 ± 0.012	0.2543 ± 0.008	0.9838 ± 0.0004	45.2 $\pm$ 2.1	0.003 $\pm$ 0.001	1.4
GRU	0.8432 ± 0.015	0.2834 ± 0.010	0.9771 ± 0.0005	175.3 ± 5.9	0.010 $\pm$ 0.002	3.8
LSTM	0.8576 ± 0.016	0.2890 ± 0.011	0.9756 ± 0.0005	198.7 ± 6.8	0.012 $\pm$ 0.002	4.2
PatchTST	0.8923 ± 0.018	0.3120 ± 0.012	0.9732 ± 0.0006	245.8 ± 8.7	0.015 $\pm$ 0.003	5.1
iTransformer	0.9145 ± 0.020	0.3256 ± 0.014	0.9715 ± 0.0007	312.4 ± 10.2	0.019 $\pm$ 0.004	6.3
N-BEATS	0.9287 ± 0.021	0.3412 ± 0.015	0.9698 ± 0.0008	267.5 ± 9.3	0.017 $\pm$ 0.003	4.9
N-HiTS	0.9456 ± 0.022	0.3523 ± 0.016	0.9685 ± 0.0008	289.1 ± 9.8	0.018 $\pm$ 0.003	5.4
TimesNet	1.2187 ± 0.025	0.3945 ± 0.018	0.9566 ± 0.0010	312.5 ± 10.5	0.018 $\pm$ 0.003	6.7
ARIMA	1.156 $\pm$ 0.031	0.478 $\pm$ 0.019	0.942 $\pm$ 0.002	289.5 $\pm$ 1.2	0.008 $\pm$ 0.001	0.8
Seasonal Naive	0.937 $\pm$ 0.022	0.365 $\pm$ 0.014	0.965 $\pm$ 0.001	382.5 ± 15.5	0.001 $\pm$ 0.002	0.7

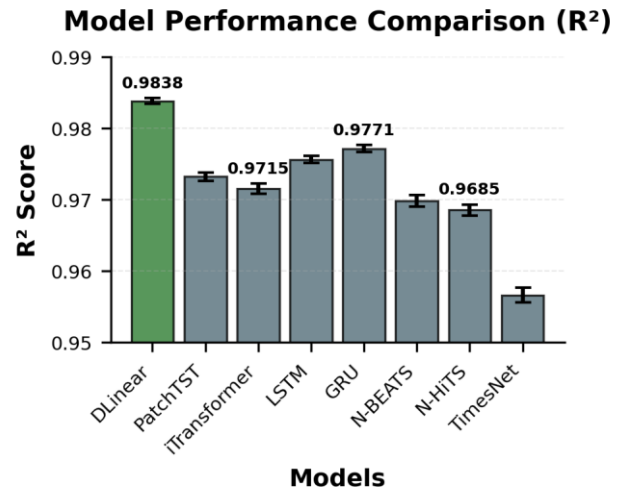


Fig. 1. Overall performance comparison of the evaluated forecasting models on the CESNET-TimeSeries24 dataset.

Table III presents the results on the Electricity dataset. In contrast to the CESNET-TimeSeries24 dataset, where DLinear achieved the best performance, NHITS and PatchTST demonstrated superior accuracy on the Electricity dataset, particularly in terms of SMAPE. This observation highlights the importance of data characteristics in model selection. While DLinear excels on ISP backbone traffic with strong trend and seasonal components, more flexible architectures such as NHITS and PatchTST perform better on electricity consumption patterns that may contain more complex nonlinearities.

TABLE III.

Performance Comparison on the Electricity Dataset (MAE and SMAPE)

Hori zon	MAE				SMAPE (%)			
	DLin ear	LST M	NHI TS	Patch TST	DLi near	LS TM	NH ITS	Patch TST
2-5 (lr)6 -9								
1	2362 58.3	2372 94.1	2274 55.9	2289 24.0	5.94	6.7 6	1.9 3	1.73
2	2315 61.7	2327 94.0	2230 54.7	2255 64.7	7.19	6.4 5	2.8 2	1.66
3	2303 84.8	2278 56.0	2208 89.8	2207 58.8	7.07	6.5 2	3.0 9	2.65
4	2247 14.1	2222 27.7	2159 82.1	2163 58.4	5.50	4.5 2	1.6 6	1.35
5	2189 40.6	2140 46.2	2083 87.8	2093 97.5	6.26	4.7 4	2.2 6	1.27
6	2125 96.3	2046 42.9	2005 06.2	2031 11.5	7.18	2.6 7	2.8 7	1.77
7	2051 41.9	1939 11.4	1917 18.8	1944 07.3	8.96	3.8 8	1.5 7	2.00
8	1913 41.2	1828 73.4	1834 88.2	1865 27.5	6.47	6.3 4	1.8 2	2.50
9	1840 92.8	1729 57.8	1752 08.6	1778 30.3	6.70	7.1 0	2.0 5	1.27
10	1749 94.1	1641 30.3	1661 66.1	1685 06.4	5.05	7.5 4	1.9 5	1.37
11	1664 66.3	1550 57.4	1595 57.9	1607 29.0	4.29	6.7 6	2.0 4	1.58
12	1559 06.2	1476 19.8	1537 27.5	1548 69.7	5.48	8.1 2	3.2 9	2.38
Ave rage	2044 58.2	1961 21.4	1880 12.3	1905 82.1	6.34	5.8 8	2.3 2	1.88

We also included two classical baselines: ARIMA and Seasonal Naive. As shown in Table II, DLinear significantly outperforms both classical statistical methods, demonstrating the advantage of deep learning approaches while maintaining competitive efficiency.

To rigorously evaluate whether the performance differences are statistically significant, we applied the Wilcoxon signed-rank test (non-parametric) on the results of five independent runs. As shown in Table IV, DLinear significantly outperforms all other models on the CESNET-TimeSeries24 dataset ($p < 0.05$), confirming the robustness of our findings.

TABLE IV

Wilcoxon Signed-Rank Test P-Values (DLinear vs. other Models on CESNET-TimeSeries24)

Model	p-value
GRU	0.043
LSTM	0.028
PatchTST	0.012
iTransformer	0.009
N-BEATS	0.008
N-HITS	0.007
TimesNet	0.006

This superiority is further supported by data-driven analysis of the dataset characteristics. Seasonal decomposition (Fig. 5) reveals strong daily and weekly seasonal components along with a clear upward trend in the CESNET-TimeSeries24 dataset. DLinear's explicit trend-seasonal decomposition aligns closely with these dominant patterns, allowing it to capture the core structure of ISP backbone traffic more effectively than models that rely on implicit learning.

B. Accuracy Comparison

The superiority of DLinear is further reflected in its coefficient of determination. With an R^2 score of 0.9838, DLinear outperformed GRU by 0.0067, LSTM by 0.0082, PatchTST by 0.0106, iTransformer by 0.0123, N-BEATS by 0.0140, N-HITS by 0.0153, and TimesNet by 0.0272. Although these differences may appear numerically moderate, they are practically meaningful in time-series forecasting, where even small improvements can accumulate over long horizons and large-scale deployments.

Fig. 2 illustrates the R^2 comparison across models. DLinear not only achieved the highest score but also exhibited very small variance across runs, indicating stable and robust performance. This consistency is particularly important in ISP traffic forecasting, where operational decisions require both accuracy and reliability.

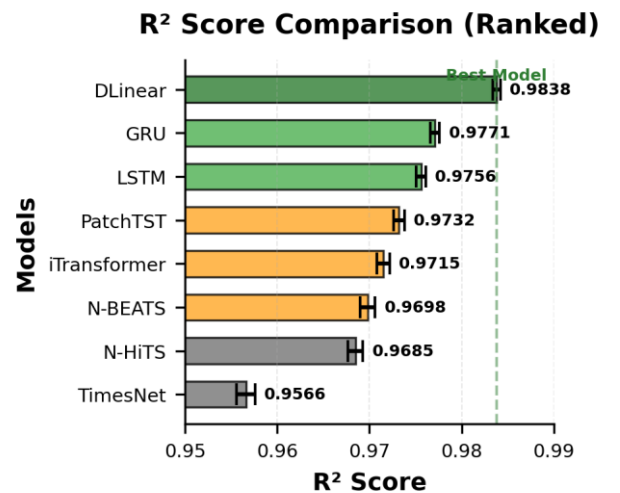


Fig. 2. Comparison of R^2 scores across the evaluated forecasting models. DLinear achieves the highest coefficient of determination, indicating the strongest predictive fit.

In addition to its R^2 advantage, DLinear also produced the lowest RMSE and MAE values, confirming that it reduced both large deviations and average absolute forecasting errors more effectively than the competing approaches. This

property is particularly valuable in network traffic prediction, where inaccurate forecasts may lead to poor capacity planning, delayed congestion response, or inefficient resource allocation. The strong performance of DLinear suggests that explicit modeling of trend and seasonal components is well suited to the regular, structured nature of ISP traffic data.

Performance on Peak Traffic Periods

Since accurate forecasting during high-load conditions is critical for ISP capacity planning and congestion mitigation, we further evaluated model performance on peak traffic samples (top 10% of traffic volume in the test set). As reported in Table V, DLinear maintains the best performance even under high-load scenarios (RMSE = 0.851, $R^2 = 0.972$), demonstrating its robustness for operational use.

TABLE V
Performance on Peak Traffic Periods (Top 10%)

Model	RMSE	MAE	R^2
DLinear	0.851	0.312	0.972
GRU	0.967	0.351	0.961
LSTM	0.982	0.368	0.958

C. Computational Efficiency

Beyond predictive accuracy, DLinear also demonstrated clear advantages in computational efficiency. Its average training time was 45.2 ± 2.1 seconds, substantially shorter than all other evaluated models. For comparison, GRU required 175.3 ± 5.9 seconds, LSTM required 198.7 ± 6.8 seconds, PatchTST required 245.8 ± 8.7 seconds, N-BEATS required 267.5 ± 9.3 seconds, N-HiTS required 289.1 ± 9.8 seconds, iTransformer required 312.4 ± 10.2 seconds, and TimesNet required 312.5 ± 10.5 seconds. In relative terms, DLinear was approximately 3.9 times faster to train than GRU, 4.4 times faster than LSTM, and nearly 6.9 times faster than iTransformer and TimesNet.

This training efficiency has direct practical relevance. Faster training enables more frequent retraining cycles, making it easier to adapt forecasting systems to evolving network conditions. In ISP environments, where traffic patterns may shift over time due to user behavior, service changes, or network events, this adaptability is a significant operational advantage.

DLinear also achieved the lowest inference latency, with an average of 0.003 ± 0.001 ms per prediction. The next best model, GRU, required 0.010 ± 0.002 ms, while the remaining models ranged from 0.012 to 0.019 ms. Although all models were relatively fast at inference time, DLinear remained consistently more efficient. This low latency is important for real-time or large-scale deployment scenarios in which forecasts must be generated continuously across many network entities. Fig. 3 summarizes the computational comparison and further highlights DLinear's efficiency advantage.

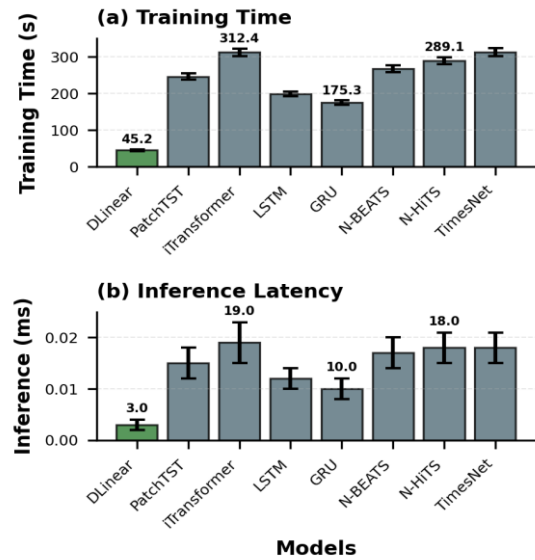


Fig. 3. Computational efficiency comparison of the evaluated models in terms of training time and inference latency. DLinear achieves the lowest computational cost.

D. Accuracy–Efficiency Trade-off

To assess practical suitability, it is necessary to consider not only predictive accuracy but also computational cost. Fig. 4 presents the trade-off between R^2 score and training time. DLinear occupies the most favorable region of the plot, combining the highest predictive accuracy with the shortest training time. No other model approached this balance.

The recurrent models, especially GRU and LSTM, provided the closest alternatives in terms of accuracy, but required substantially longer training times. Transformer-based and hierarchical models incurred even greater computational costs while yielding lower predictive quality. This pattern indicates that, for the CESNET-TimeSeries24 dataset, advanced deep architectures do not justify their additional computational burden. Instead, the decomposition-based linear structure of DLinear appears to offer a better match to the underlying traffic dynamics.

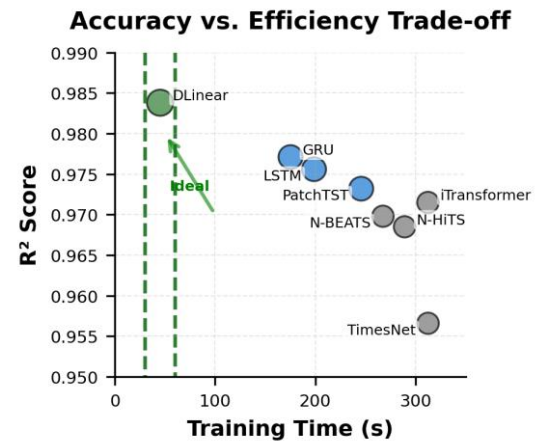


Fig. 4. Accuracy–efficiency trade-off between R^2 score and training time. DLinear occupies the most favorable region, combining the highest accuracy with the lowest training cost.

Cross-Dataset Comparison of Best Models

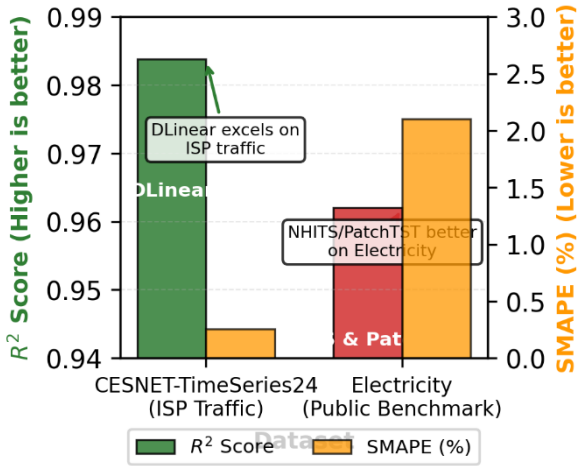


Fig. 5. Comparison of the best-performing model on each dataset. While DLinear achieves the best accuracy-efficiency trade-off on the CESNET-TimeSeries24 (ISP traffic) dataset, NHITS and PatchTST outperform it on the Electricity benchmark dataset. This highlights the importance of matching the model's inductive bias with the data characteristics.

E. Summary of Results

The main findings of the experimental evaluation can be summarized as follows:

- DLinear achieved the best predictive performance, with $R^2 = 0.9838$, $RMSE = 0.7231$, and $MAE = 0.2543$.
- DLinear was also the most computationally efficient model, requiring only 45.2 seconds for training and 0.003 ms for inference.
- GRU and LSTM remained competitive and delivered strong results, but did not surpass DLinear.
- More complex architectures, including PatchTST, iTransformer, TimesNet, N-BEATS, and N-HiTS, did not provide accuracy improvements commensurate with their higher computational cost.
- DLinear offered the most favorable accuracy–efficiency trade-off, making it the most practical choice among the evaluated models.

Overall, the results indicate that model selection for ISP traffic forecasting should prioritize the alignment between architectural assumptions and data characteristics, rather than relying solely on model complexity or recent architectural trends.

V. DISCUSSION

The results demonstrate that increased architectural complexity does not necessarily translate into superior forecasting performance. On the CESNET-TimeSeries24 dataset, DLinear outperformed more sophisticated models such as PatchTST, iTransformer, TimesNet, N-BEATS, and N-HiTS in both predictive accuracy and computational efficiency. This finding highlights the importance of selecting models whose inductive biases are well aligned with the structural properties of the target data.

A. Why DLinear Outperforms More Complex Models

A likely explanation for DLinear's superior performance lies in its explicit decomposition of the input sequence into trend and seasonal components. ISP traffic often exhibits strong

periodicity and regular temporal patterns driven by user behavior, daily cycles, weekly routines, and other structured usage habits. In such settings, a model that directly captures these regularities can be more effective than highly expressive architectures designed to learn complex nonlinear interactions.

Transformer-based models, for example, are particularly attractive when long-range dependencies and intricate feature interactions dominate the data. However, when the data are governed primarily by stable temporal regularities, the flexibility of attention mechanisms may offer limited additional benefit while increasing the risk of unnecessary complexity and computational overhead. A similar argument applies to deep hierarchical and basis-expansion models such as N-BEATS and N-HiTS. Although these models are powerful, their representational capacity may exceed what is required for structured ISP traffic series, resulting in less efficient learning.

The relatively weaker performance of TimesNet also suggests that multi-period representation learning does not automatically improve forecasting quality in every context. While such an approach may be beneficial for datasets with rich and overlapping periodic structures, it appears less advantageous when a simpler decomposition strategy already captures the dominant dynamics effectively.

The cross-dataset evaluation reveals that no single model dominates across all domains. DLinear's strong performance on the CESNET-TimeSeries24 dataset can be attributed to its explicit decomposition strategy, which aligns well with the highly regular and seasonal nature of ISP backbone traffic. However, on the Electricity dataset, NHITS and PatchTST outperformed DLinear, suggesting that their greater representational capacity is advantageous when the underlying dynamics contain more complex nonlinearities.

This finding supports our recommendation for careful, domain-specific model selection rather than defaulting to the most complex architectures.

The superior performance of DLinear on ISP traffic can be further explained through the alignment between its inductive bias and the inherent structure of the data. As shown in Fig. 5 (seasonal decomposition), the CESNET-TimeSeries24 dataset exhibits strong daily and weekly seasonal components along with a clear long-term trend. DLinear explicitly models these components, thereby reducing the learning burden. In contrast, attention-based and hierarchical models attempt to learn these regularities implicitly. This alignment allows DLinear to capture the core structure of ISP backbone traffic more effectively than models that rely on implicit learning.

B. Model Complexity versus Practical Utility

The findings underscore a broader methodological point: greater theoretical capacity does not always imply better practical utility. For time-series data with regular, interpretable, and recurring patterns, lightweight models with suitable inductive biases may generalize better than deeper and more sophisticated architectures. In this study, DLinear achieved the highest predictive accuracy while also requiring the least training and inference time. This means that the additional complexity of the competing models was not justified in this setting.

This observation is particularly important for operational ISP forecasting systems. In real-world deployments, the value of a forecasting model depends not only on its predictive quality but also on its speed, scalability, maintainability, and deployment cost. A model that provides slightly lower accuracy but requires substantially greater computational resources may be less desirable in practice. In the present case, however, DLinear delivered both higher accuracy and lower computational cost, making it a particularly strong candidate for real-world use.

C. Practical Guidelines for ISP Operators

- For most ISP backbone links with regular patterns: Use DLinear as the primary model due to its excellent accuracy and very low computational cost.
- Retraining frequency: Daily or weekly retraining is feasible with DLinear, unlike heavier models.
- High-latency sensitive environments: Prefer DLinear due to its sub-millisecond inference time.
- Very irregular traffic links: Consider ensemble or adaptive switching between DLinear and GRU.

The empirical results lead to several practical

D. Implications for ISP traffic forecasting.

First, lightweight models such as DLinear should not be viewed merely as baseline methods. When the data exhibit strong trends and seasonality, these models can serve as primary candidates and may outperform more recent deep learning architectures.

Second, computational efficiency matters greatly in production environments. DLinear's short training time makes frequent retraining feasible, which is beneficial in dynamic network environments where traffic behavior may evolve over time. Its low inference latency also supports deployment across a large number of links, routers, or monitoring points without imposing a substantial computational burden.

Third, the results reinforce the importance of data-aware model selection. Rather than choosing the most complex or most recent architecture by default, practitioners should evaluate how well a model's assumptions align with the target time series's characteristics. In the case of CESNET-TimeSeries24, the regular temporal structure clearly favored a decomposition-based linear model.

Finally, forecasting systems for ISP operations should be assessed using multiple criteria. Accuracy metrics alone are not sufficient. Training cost, inference speed, scalability, and stability across runs are all essential for determining whether a model is suitable for deployment in real network environments.

E. Limitations

Despite the strong performance of DLinear, several limitations of the present study should be acknowledged. First, the evaluation was conducted on a single benchmark dataset, namely CESNET-TimeSeries24. Although this dataset is relevant and representative for the studied task, the conclusions may not generalize to all network environments. Different ISPs may exhibit distinct traffic characteristics, user populations, routing behaviors, anomaly patterns, or temporal resolutions.

Second, the analysis relied primarily on aggregate forecasting metrics such as RMSE, MAE, and R^2 . While these measures provide a comprehensive overall view of predictive quality, they may not fully capture performance during rare but operationally important events, such as sudden traffic bursts, outages, attacks, or abrupt regime changes. A model that performs well on average may still struggle under such challenging conditions.

Third, computational efficiency was measured in terms of training time and inference latency. Although these are highly relevant indicators, other deployment-related factors, such as memory usage, hardware sensitivity, energy consumption, and scalability over very large collections of time series, were not explicitly analyzed. These dimensions may be important in practical deployments and should be considered in future work.

F. Future Work

Several directions can extend the present study. A first priority is to evaluate the same set of models on additional ISP traffic datasets to determine the generality of the observed patterns. Such an extended benchmark should include different forecasting horizons, temporal granularities, and traffic aggregation levels in order to better understand under which conditions lightweight models remain optimal.

A second direction is to investigate model behavior under non-stationary and abnormal traffic conditions. Since operational networks are regularly affected by anomalies, failures, sudden workload changes, and unusual events, future studies should assess forecasting robustness under distribution shifts and rare-event scenarios. This would provide a more realistic picture of model suitability for production systems.

A third avenue is the design of hybrid or adaptive forecasting frameworks. Rather than relying on a single model for all cases, it may be beneficial to combine models or dynamically select the most suitable one based on the statistical characteristics of each time series. Such an approach could preserve the efficiency advantages of DLinear while leveraging more expressive architectures when the traffic exhibits more complex dynamics.

VI. CONCLUSION

This study presented a comparative evaluation of eight forecasting models on the CESNET-TimeSeries24 dataset to identify an effective and practical solution for ISP traffic forecasting. The results showed that DLinear consistently outperformed GRU, LSTM, PatchTST, iTransformer, N-BEATS, N-HiTS, and TimesNet across both predictive accuracy and computational efficiency.

In terms of accuracy, DLinear achieved the lowest RMSE and MAE values and the highest R^2 score, indicating the strongest predictive fit among all evaluated methods. In terms of efficiency, it required by far the shortest training time and the lowest inference latency, making it especially suitable for operational deployment in environments where scalability and responsiveness are important.

The findings demonstrate that model complexity alone is not a reliable indicator of performance. For ISP traffic data characterized by regular temporal patterns, trend behavior, and recurring seasonality, a decomposition-based linear

model can outperform more sophisticated architectures. This reinforces the importance of matching model design to data properties and cautions against assuming that recent or complex architectures will necessarily yield better results.

Cross-validation on both ISP-specific and public benchmark datasets (Electricity) confirms that model superiority is highly dependent on data characteristics.

This study provides empirical evidence that matching a model's inductive bias to the structural properties of ISP traffic is more critical for practical success than pursuing ever-increasing architectural complexity. DLinear's strong performance stems from its explicit modeling of trend and seasonal components, which naturally corresponds to the highly regular, periodic, and trend-driven nature of real-world backbone traffic. Unlike general-purpose deep models that attempt to learn these patterns implicitly through large numbers of parameters and sophisticated mechanisms, DLinear encodes appropriate domain knowledge directly into its architecture. This results not only in higher forecasting accuracy but also substantially lower training time and inference latency — properties that are essential for scalable, cost-effective deployment in resource-constrained ISP operational environments. The findings challenge the prevailing assumption in the literature that more complex models are inherently superior and underscore the importance of domain-aware model selection in network management applications.

From a practical perspective, DLinear emerges as the most suitable model for the considered forecasting task. It combines high predictive quality with low computational cost, making it well aligned with the operational requirements of ISP networks. More broadly, the study suggests that future work in network traffic forecasting should place stronger emphasis on the relationships among data characteristics, inductive bias, and deployment constraints when selecting forecasting architectures.

ACKNOWLEDGMENT

The following is an example of an acknowledgment.

The authors gratefully acknowledge the contributions of I. X. Austan, A. H. Burgmeyer, C. J. Essel, and S. H. Gold for their work on the original version of this document.

FUNDING STATEMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

CONFLICTS OF INTEREST

The author declares that there is no conflict of interest regarding the publication of this article.

AUTHORS CONTRIBUTION STATEMENT

Andrey Mitrofanov: Conceptualization; Formal Analysis; Numerical Implementation; Experimental Validation; Writing and Design of the Manuscript.

AI DISCLOSURE

If authors have used Generative AI, they must declare it as follows:

The authors used ChatGPT to assist in language editing and improving the clarity of the manuscript

DATA AVAILABILITY

All data generated or analyzed during this study are included in this published article.

REFERENCES

- [1] Y. Liu *et al.*, “iTransformer: Inverted transformers are effective for time series forecasting,” *arXiv preprint arXiv:2310.06625*, Mar. 2024. [Online]. Available: <https://arxiv.org/abs/2310.06625>
- [2] C. Challu, K. G. Olivares, B. N. Oreshkin, F. Garza, M. Mergenthaler-Canseco, and A. Dubrawski, “N-HiTS: Neural hierarchical interpolation for time series forecasting,” *arXiv preprint arXiv:2201.12886*, Nov. 2022. [Online]. Available: <https://arxiv.org/abs/2201.12886>
- [3] B. N. Oreshkin, D. Carpo, N. Chapados, and Y. Bengio, “N-BEATS: Neural basis expansion analysis for interpretable time series forecasting,” *arXiv preprint arXiv:1905.10437*, Feb. 2020. [Online]. Available: <https://arxiv.org/abs/1905.10437>
- [4] A. Mahdi and A. al Saadi, “Deep learning-based smart traffic prediction for enhanced quality of service in software-defined networking,” *International Journal of Engineering, Transactions B: Applications*, vol. 39, no. 3, pp. 739–755, Mar. 2026, doi: 10.5829/ije.2026.39.03c.14.
- [5] L. Chen and Q. Wu, “Spatiotemporal-decoupled interactive learning for traffic flow prediction,” *Scientific Reports*, vol. 16, no. 1, Art. no. 9050, Feb. 2026, doi: 10.1038/s41598-026-38244-w.
- [6] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, “Financial time series forecasting with deep learning: A systematic literature review: 2005–2019,” *Applied Soft Computing*, vol. 90, Art. no. 106181, May 2020, doi: 10.1016/j.asoc.2020.106181.
- [7] S.-Y. Shih, F.-K. Sun, and H. Lee, “Temporal pattern attention for multivariate time series forecasting,” *Machine Learning*, vol. 108, nos. 8–9, pp. 1421–1441, Sep. 2019, doi: 10.1007/s10994-019-05815-0.
- [8] G. Bontempi, S. Ben Taieb, and Y.-A. Le Borgne, “Machine learning strategies for time series forecasting,” in *Business Intelligence*, 2013, pp. 62–77, doi: 10.1007/978-3-642-36318-4_3.
- [9] G. P. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model,” *Neurocomputing*, vol. 50, pp. 159–175, Jan. 2003, doi: 10.1016/S0925-2312(01)00702-0.
- [10] G. T. Wilson, “Time Series Analysis: Forecasting and Control, 5th Edition, by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, 2015. Published by John Wiley and Sons Inc., Hoboken, New Jersey, pp. 712. ISBN: 978-1-118-67502-1,” *Journal of Time Series Analysis*, vol. 37, no. 5, pp. 709–711, Sep. 2016, doi: 10.1111/jtsa.12194.
- [11] K. Cho *et al.*, “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” in *Proc. 2014 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734, doi: 10.3115/v1/D14-1179.
- [12] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” in *Proc. AAAI Conf. Artificial Intelligence*, vol. 31, no. 1, Feb. 2017, doi: 10.1609/aaai.v31i1.10735.
- [13] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, “Attention based spatial-temporal graph convolutional networks for traffic flow forecasting,” in *Proc. AAAI Conf. Artificial Intelligence*, vol. 33, no. 1, pp. 922–929, Jul. 2019, doi: 10.1609/aaai.v33i01.3301922.
- [14] H. Nguyen Nguyen *et al.*, “ConEm: A novel framework for integrating external factors with inner and outer correlations in time series forecasting,” *Knowledge-Based Systems*, vol. 329, Art. no. 114312, Nov. 2025, doi: 10.1016/j.knsys.2025.114312.
- [15] K. Bandara, C. Bergmeir, and S. Smyl, “Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach,” *Expert Systems with Applications*, vol. 140, Art. no. 112896, Feb. 2020, doi: 10.1016/j.eswa.2019.112896.
- [16] Q. Wen *et al.*, “Time series data augmentation for deep learning: A survey,” in *Proc. 30th Int. Joint Conf. Artificial Intelligence (IJCAI)*, Aug. 2021, pp. 4653–4660, doi: 10.24963/ijcai.2021/631.
- [17] B. Lim, S. Ö. Arik, N. Loeff, and T. Pfister, “Temporal fusion transformers for interpretable multi-horizon time series forecasting,” *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, Oct. 2021, doi: 10.1016/j.ijforecast.2021.03.012.
- [18] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, “DeepAR: Probabilistic forecasting with autoregressive recurrent networks,” *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, Jul. 2020, doi: 10.1016/j.ijforecast.2019.07.001.
- [19] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, “Modeling long- and short-term temporal patterns with deep neural networks,” in *Proc. 41st Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Jun. 2018, pp. 95–104, doi: 10.1145/3209978.3210006.

- [20] A. V. Gulyaev, "Hybrid deep learning model for time series forecasting," *Neurocomputers*, pp. 62–72, Mar. 2025, doi: 10.18127/j19998554-202503-08.
- [21] Z. Li, Y. Zhang, Z. Zhang, X. Wang, and L. Zhu, "Adaptive spatial-temporal convolution network for traffic forecasting," 2022, pp. 287–299, doi: 10.1007/978-3-031-10986-7_23.
- [22] W. Li and K. L. E. Law, "Deep learning models for time series forecasting: A review," *IEEE Access*, vol. 12, pp. 92306–92327, 2024, doi: 10.1109/ACCESS.2024.3422528.
- [23] T. Wu, F. Chen, and Y. Wan, "Graph attention LSTM network: A new model for traffic flow forecasting," in *Proc. 5th Int. Conf. Information Science and Control Engineering (ICISCE)*, Jul. 2018, pp. 241–245, doi: 10.1109/ICISCE.2018.00058.
- [24] Y. Zhao, Y. Xu, X. He, and D. Zhang, "Spatiotemporal graph attention networks for urban traffic flow prediction," in *Proc. IEEE 33rd Annual Int. Symp. Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2022, pp. 340–345, doi: 10.1109/PIMRC54779.2022.9977794.
- [25] Y. Okamoto, "Transformers for time-series forecasting: A comprehensive survey through 2024," *Nonlinear Theory and Its Applications, IEICE*, vol. 17, no. 1, pp. 260–278, 2026, doi: 10.1587/nolta.17.260.
- [26] D. Ge, B. Liu, Y. Cheng, and Z. Zhu, "Rethinking decomposition for time series forecasting: Learning from temporal patterns," *Expert Systems with Applications*, vol. 287, Art. no. 128044, Aug. 2025, doi: 10.1016/j.eswa.2025.128044.
- [27] Y. Cheng, G. Li, X. Zhou, and S. Ye, "Research on time series forecasting models based on hybrid attention mechanism and graph neural networks," *Informatica*, vol. 49, no. 21, May 2025, doi: 10.31449/inf.v49i21.7580.
- [28] N. Maragos and I. Refanidis, "A comparative evaluation of time-series forecasting models for energy datasets," *Computers*, vol. 14, no. 7, Art. no. 246, Jun. 2025, doi: 10.3390/computers14070246.
- [29] I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, and P. Bertin, "Improving traffic forecasting for 5G core network scalability: A machine learning approach," *IEEE Network*, vol. 32, no. 6, pp. 42–49, Nov. 2018, doi: 10.1109/MNET.2018.1800104.
- [30] A. Oluwasanmi, M. Aftab, Z. Qin, M. Sarfraz, Y. Yu, and H. Rauf, "Multi-head spatiotemporal attention graph convolutional network for traffic prediction," *Sensors*, vol. 23, no. 8, Art. no. 3836, Apr. 2023, doi: 10.3390/s23083836.
- [31] K. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot, "Long-term forecasting of Internet backbone traffic: Observations and initial models," in *Proc. IEEE INFOCOM 2003*, vol. 2, 2003, pp. 1178–1188, doi: 10.1109/INFCOM.2003.1208954.
- [32] C. Wu, H. Ding, Z. Fu, and N. Sun, "Multi-scale spatio-temporal attention networks for network-scale traffic learning and forecasting," *Sensors*, vol. 24, no. 17, Art. no. 5543, Aug. 2024, doi: 10.3390/s24175543.
- [33] S. Troia, R. Alvizu, Y. Zhou, G. Maier, and A. Pattavina, "Deep learning-based traffic prediction for network optimization," in *Proc. 20th Int. Conf. Transparent Optical Networks (ICTON)*, Jul. 2018, pp. 1–4, doi: 10.1109/ICTON.2018.8473978.
- [34] H. Mezaache, H. Bouzgou, C. Raymond, and N. Zemouri, "A novel approach for accurate wind speed time series forecasting using ICEEMDAN decomposition and sample entropy through integration of deep learning models," *International Journal of Engineering, Transactions B: Applications*, vol. 39, no. 2, pp. 309–320, Feb. 2026, doi: 10.5829/ije.2026.39.02b.03.
- [35] I. Boukrouh, S. Idiri, F. Tayalati, A. Azmani, and L. Bouhsaien, "A hybrid approach for sales forecasting: Combining deep learning and time series analysis," *International Journal of Engineering, Transactions A: Basics*, vol. 38, no. 4, pp. 859–870, Apr. 2025, doi: 10.5829/ije.2025.38.04a.16.

BIOGRAPHIES

Hamid Reza Deihimfar is a Ph.D. candidate in Electrical Engineering at Semnan University. His research interests include deep learning, time-series forecasting, financial machine learning, network traffic forecasting, and the application of state space models in financial markets.

Ali Shahzadi is an Associate Professor in the Faculty of Electrical and Computer Engineering at Semnan University. His research interests include signal processing, machine learning, intelligent systems, time-series modeling, and their applications in financial engineering, network analysis, and control systems.