MSTN: A Lightweight and Fast Model for General Time-Series Analysis

Anonymous authors
Paper under double-blind review

Abstract

Real-world time series often exhibit strong non-stationarity, complex nonlinear dynamics, and behavior expressed across multiple temporal scales, from rapid local fluctuations to slow-evolving long-range trends. However, many contemporary architectures impose rigid, fixed-scale structural priors—such as patch-based tokenization, predefined receptive fields, or frozen backbone encoders—which can over-regularize temporal dynamics and limit adaptability to abrupt high-magnitude events. To handle this, we introduce the *Multi-scale Temporal Network* (MSTN), a hybrid neural architecture grounded in an *Early Temporal Aggregation* principle. MSTN integrates three complementary components: (i) a multi-scale convolutional encoder that captures fine-grained local structure; (ii) a sequence modeling module that learns long-range dependencies through either recurrent or attention-based mechanisms; and (iii) a self-gated fusion stage incorporating squeeze–excitation and multi-head attention to dynamically modulate cross-scale representations. This design enables MSTN to flexibly model temporal patterns spanning milliseconds to extended horizons, while avoiding the computational burden typically associated with long-context models. Across extensive benchmarks covering forecasting, imputation, classification, and cross-dataset generalization, MSTN achieves state-of-the-art performance, establishing new best results on **24 of 32** datasets, while remaining lightweight **($\approx$ 1M params)** and suitable for low-latency **(<1 sec, often in milliseconds)**, resource-constrained deployment.

1 Introduction

Multivariate time series (MTS) analysis underpins a broad spectrum of societally and industrially critical applications, including healthcare monitoring Morid et al. (2023), intelligent transportation systems Kadiyala & Kumar (2014), climate and environmental modeling Gruca et al. (2022), energy systems Kardakos et al. (2013), industrial prognostics, and behavioural analytics. Across these domains, learning from temporal data is central to tasks such as long-term forecasting, missing-value imputation, and trajectory or activity classification Wu et al. (2022); Zhao et al. (2024). Despite decades of study, robust temporal modeling remains fundamentally challenging. Unlike language or vision, individual time points in a time series carry limited semantic meaning; useful information emerges only through temporal variation patterns such as continuity, periodicity, and long-term trends Wu et al. (2023); Lim & Zohren (2021). This intrinsic dependency structure renders time series modeling both information-rich and computationally demanding.

Deep learning (DL) has substantially advanced time series analysis, not only improving predictive accuracy but also enabling transferable representations for downstream tasks Nie et al. (2023). However, recent evidence that simple linear models can rival or even surpass sophisticated Transformer-based architectures on standard benchmarks Zeng et al. (2022) has exposed structural limitations in contemporary designs. Classical neural architectures—including convolutional networks Franceschi et al. (2019); Bai et al. (2018) and recurrent models with gating mechanisms Hochreiter & Schmidhuber (1997); Lai et al. (2018)—are constrained by vanishing gradients, limited parallelism, or restricted receptive fields. Although temporal convolutional networks improve efficiency He & Zhao (2019), they rely on fixed receptive scales. Transformer-based models promise global dependency modeling Vaswani et al. (2023), yet their quadratic complexity in sequence length has driven a proliferation of sparsification and approximation strategies Zhou et al. (2021;

2022a). Many such methods continue to treat time steps as independent tokens, limiting their ability to encode local semantic structure and multi-scale temporal interactions.

These design pressures have led to the emergence of three dominant modeling paradigms, formalized as Channel Strategies Qiu et al. (2025): Channel Independence (CI), Channel Dependence (CD), and Channel Partiality (CP). CI-based approaches prioritize computational efficiency by processing each variable independently, but sacrifice inter-variable interactions. CD-based models explicitly capture cross-channel dependencies at the cost of poor scalability. CP-based methods seek a compromise by projecting channels into lower-dimensional latent spaces, improving efficiency while retaining partial correlations. Despite their differences, all three paradigms rely on rigid structural assumptions—fixed patch sizes, predefined resolutions, uniform-scale mixing MLPs, static backbones, or explicit periodic decompositions—which limit adaptability to non-stationary, irregular, and long-sequence dynamics. As a result, even state-of-the-art (SOTA) systems such as PatchTST Nie et al. (2023), TimesNet Wu et al. (2023), SOFTS Han et al. (2024a), MCformer Han et al. (2024b), TSMixer Chen et al. (2023), and recent LLM-inspired approaches Chang et al. (2025); Jin et al. (2024) struggle to maintain temporal fidelity under complex, real-world conditions.

A key limitation shared by these approaches is the implicit assumption of a single dominant temporal scale. This limitation is architecturally evident in designs such as TSMixer, where the time-mixing MLPs are shared across all time steps, resulting in a fixed-resolution, uniform transformation across the temporal dimension. Real-world temporal processes, however, are inherently multi-scale: micro-level fluctuations, sudden spikes, intermediate dynamics, and long-term trends often coexist and interact. Capturing such a hierarchical structure is essential in safety-critical and high-variability domains, including physiological monitoring, risky driving behavior analysis, and mechanical prognostics. Consequently, fixed-scale or uniform-scale architectures are ill-suited to these settings. Existing models handle a single task or focus on a single type of temporal variations- short or long, rarely balancing both effectively. There is a need for architectures that are not only general-purpose but also lightweight and efficient.

To address these challenges, we introduce the *Multi-scale Temporal Network* (MSTN), a hybrid architecture grounded in the principle of *Early Temporal Aggregation* (ETA). The central idea of ETA is to decouple expressive temporal modeling from inference-time complexity by collapsing the temporal dimension early in the network via an $L \to 1$ transformation, where L denotes the lookback-window length. By performing the computationally intensive operations (e.g., $\mathcal{O}(L)$ BiLSTM or $\mathcal{O}(L^2)$ Transformer self-attention) *before* this aggregation step, the subsequent refinement layers—including feature fusion, SE recalibration, MHA, and prediction modules—operate with a fixed $\mathcal{O}(1)$ cost with respect to L. MSTN achieves this through a dual-path encoder that integrates complementary temporal strategies. Multi-scale convolutional encoders capture fine-grained local patterns, while a sequence modeling backbone (BiLSTM or Transformer) captures long-range dependencies. Their outputs are fused through a self-gated fusion (SGF) mechanism, augmented with squeeze-and-excitation (SE) recalibration and multi-head attention (MHA), producing compact, discriminative, and temporally coherent representations.

We evaluate MSTN across forecasting, imputation, classification, and cross-domain generalization tasks, demonstrating consistent SOTA performance on 24 of 32 benchmark datasets. MSTN establishes new best results in long-term forecasting, achieves strong robustness under high missingness in imputation, and delivers superior classification accuracy across diverse application domains. In particular, these gains are achieved alongside efficiency improvements: the MSTN-BiLSTM and MSTN-Transformer variants utilize fixed cores of ∼0.40 million and ∼1.04 million parameters, respectively. The proposed work thus has a twofold impact. First, MSTN provides fast inference and can be embedded in real-time applications for time-series classification/forecasting, such as healthcare monitoring, industrial machine health, flight navigation, etc. Second, MSTN has a very small memory footprint and is thus suited for edge-device applications.

The key contributions of this work are summarized as follows:

1. We introduce MSTN, a multi-scale neural framework that is both fast and efficient. By leveraging the ETA principle to confine expensive $O(L^2)$ operations to the initial encoder, MSTN achieves a constant $\boldsymbol{O(1)}$ complexity with respect to the sequence length. MSTN incurs ∼**0.40M** (MSTN-BiLSTM) and ∼**1.04M** (MSTN-Transformer) parameters and **sub-millisecond inference** latency.

2. Through an extensive evaluation across 32 standard benchmarks spanning imputation, forecasting, classification, and cross-dataset generalization, MSTN achieves **SOTA performance on 24 datasets,** outperforming recent models such as EMTSF, DARTS-TS, LLM4TS, TimesNet, and PatchTST.

3. We also show the generalizability of MSTN on **7 datasets** from various domains such as health, safety, etc.

2 Related Work

In multivariate time series, a lot of work has been done. However, we are going to discuss only the most recent work in the literature of multivariate time series. Our discussion focuses on two aspects of time-series models: (i) the tasks they handle and (ii) efficiency.

2.1 Task Perspective

Work in time-series prediction can be categorized based on the following tasks: Imputation, long-term forecasting, classification, and anomaly detection. In this section, we focus on the first three tasks.

The majority of works focus on long-term forecasting tasks, including specialized architectures — such as DLinear Zeng et al. (2022), PatchTST Nie et al. (2023),iTransformer Liu et al. (2024), SOFTS Han et al. (2024a), TIME-LLM Jin et al. (2024), LLM4TS Chang et al. (2025)), and masked autoencoders (HiMTM Zhao et al. (2024)). Seg-MoE Ortigossa & Segal (2026) improves scalability and long-term dependency modeling via segment-wise Mixture-of-Experts routing, but is primarily designed for forecasting and does not explicitly handle irregular or event-driven patterns. vLinear Yue et al. (2026) introduces an efficient linear multivariate forecaster, achieving strong performance; however, its linear formulation limits its capacity to capture complex nonlinear temporal dynamics. These models have driven significant performance gains, but have not been applied to other time-series tasks like imputation or classification, which we address in the present work. In addition, the aforementioned works claim to handle channel dependency (CI, CD, or CP) differently. CI works like DLinear Zeng et al. (2022) and PatchTST Nie et al. (2023) handle time-series where channels are independent and therefore cannot model cross-channel effects. CD models such as iTransformer Liu et al. (2024), graph-based approaches such as TPGNN Liu et al. (2022) capture full cross-channel dependencies but scale poorly with channel count. CP methods like MCformer Han et al. (2024b) and ModernTCN donghao & wang xue (2024) project the channel dimension into a latent space before interaction, trading expressivity for efficiency. While these methods show impressive performance, they are limited in capturing the multi-scale temporal dynamics within and across channels—a capability essential for modeling long-sequence dependencies. To combine the strengths of both paradigms, our work takes an intermediate route where we leverage both CI and CD.

Another line of work in time-series prediction is imputation, which addresses the challenge of reconstructing missing values caused by sensor failures or data corruption, an essential step to maintain data integrity in real-world systems Zhou et al. (2023); Wu et al. (2023). Imputation tasks employ specialized architectures such as TimesNet Wu et al. (2023), GPT2(3) Zhou et al. (2023), PatchTST Nie et al. (2023), LightTS Zhang et al. (2022), and DLinear Zeng et al. (2022). However, these methods perform poorly on aperiodic or irregularly sampled events (e.g., emergency braking) because they have not been designed to detect such chaotic events. On the other hand, in the present work, MSTN can handle such events gracefully.

Another time series task, classification, involves categorizing the entire sequence to identify underlying events or patterns, enabling automated diagnosis and monitoring in fields such as healthcare and industrial maintenance Wu et al. (2023); Zhang et al. (2022). Classification specialized architectures such as GPT2(6) Alharthi et al. (2025), TimesNet Wu et al. (2023), DLinear Zeng et al. (2022), ETSFormer Woo et al. (2022), FEDformer Zhou et al. (2022a), and Informer Zhou et al. (2021). However, by design, these methods focus on seasonal decomposition, which can lead to over smoothing of temporal dynamics. Consequently, they often miss sudden, aperiodic anomalies such as spikes or crashes. They also tend to specialize in either local or global trends, rarely balancing both effectively.

2.2 Efficiency Perspective

In this section, we discuss efficiency of the time series models in terms of (i) characteristics of time series captured such as periodicity, varying temporal patterns, non-stationary behaviors, etc.; (ii) inference latency: and (iii) memory.

Periodic models such as TimesNet Wu et al. (2023), Autoformer Wu et al. (2022), ETSFormer Woo et al. (2022), FEDformer Zhou et al. (2022a), LightTS Zhang et al. (2022), PatchTST Nie et al. (2023), and DLinear Zeng et al. (2022) explicitly model periodic or seasonal structures and have demonstrated strong performance on long-term forecasting tasks. However, they often struggle to capture aperiodic, irregular, sudden spikes or event-driven dynamics. Dilated convolutional networks (e.g., ModernTCN donghao & wang xue (2024)) extract multiscale local patterns but have limited receptive fields for very long dependencies. Graph-based methods such as GTS Shang et al. (2021) and FourierGNN Yi et al. (2023) model spatio-temporal relationships but often assume static graphs, struggling with dynamically changing sensor relationships. Efficient attention variants (Informer Zhou et al. (2021), FEDformer Zhou et al. (2022a), Autoformer Wu et al. (2022)) use sparse or frequency-domain attention for long-range modeling. Despite these advances, these methods maintain the full sequence length throughout the network, incurring substantial memory and computational cost as the length of the sequence increases. They also tend to specialize in either local or global trends, rarely balancing both effectively while meeting the requirement for low-latency inference.

Recent MLP-based architectures such as TSMixer Chen et al. (2023) and TTM Ekambaram et al. (2024) achieve computational efficiency through temporal and channel mixing, demonstrating competitive performance with significantly lower computational costs compared to attention-based models. However, these architectures employ fixed mixing operations that may not adapt dynamically to varying temporal patterns across different scales and frequencies. A distinct line of work employs large language models (LLM) for time series (e.g., TIME-LLM Jin et al. (2024), LLM4TS Chang et al. (2025)), leveraging their few-shot capabilities for cold-start scenarios. However, these models suffer from high latency and memory footprints, making them unsuitable for real-time edge deployment. Other methods, such as SOFTS Han et al. (2024a), use stochastic pooling to reduce resolution but do not fully decouple sequence length from downstream processing, incur high inference latency and parameter counts. There is a need for architectures that are not only general-purpose but also lightweight and efficient. Our work attempts to address this gap.

3 Methodology

This section details the MSTN architecture and its methodological innovations, followed by the baseline models, and training configurations.

3.1 Proposed Architecture: MSTN

3.1.1 Architecture Overview

The proposed *Multi-scale Temporal Network* (MSTN; Fig. 1) is a hybrid deep-learning architecture designed around a parallel multi-scale modeling paradigm. MSTN processes temporal signals through two coordinated encoding branches, each specialized for complementary aspects of temporal structure. A key design element is the use of an *early temporal aggregation* (ETA) mechanism, applied directly to the outputs of the dual encoders, which collapses the full input sequence of length L into a single fixed-dimensional representation ($L \to 1$). This operation ensures that all subsequent refinement stages operate in constant time with respect to sequence length, thereby enabling efficient processing of long temporal inputs.

The network comprises three principal modules. (i) A *multi-scale convolutional branch* captures localized temporal patterns using hierarchically stacked convolutions; its output is aggregated via global average pooling, promoting robustness and translation invariance. (ii) A *sequence modeling branch* captures long-range dependencies, instantiated using either a bidirectional LSTM (MSTN-BiLSTM) or a Transformer encoder (MSTN-Transformer). To maintain computational efficiency, this branch is compressed through sequence mean pooling, yielding a global summary vector. (iii) A *multi-scale fusion and refinement module* integrates

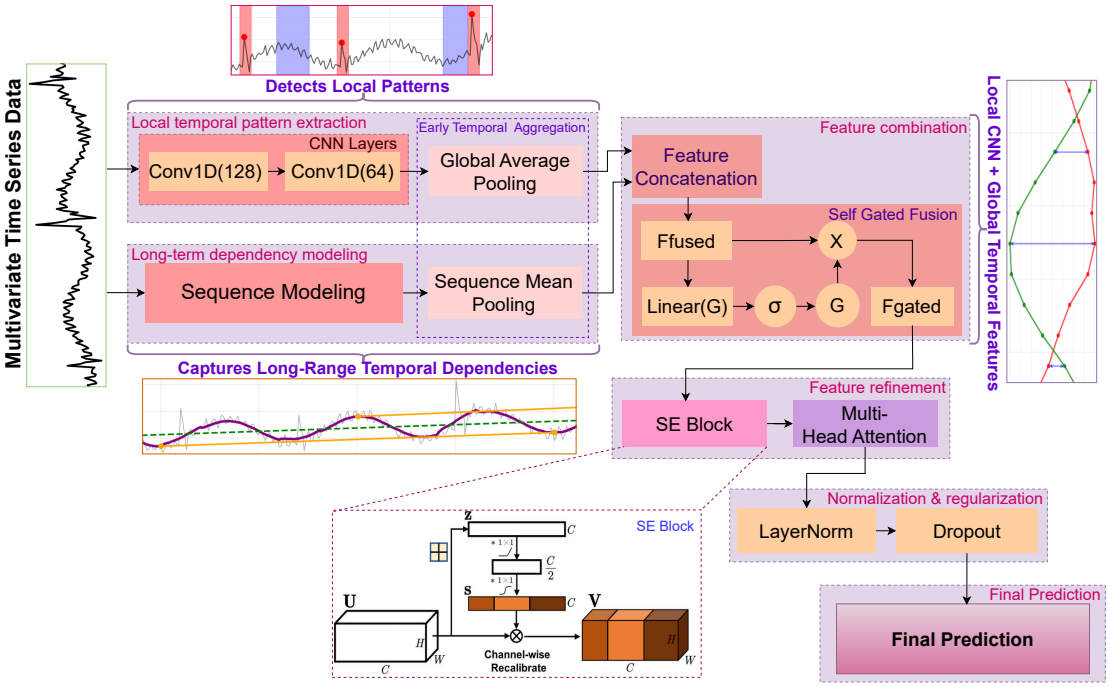

Figure 1: Architecture of the proposed Multi-scale Temporal Network (MSTN).

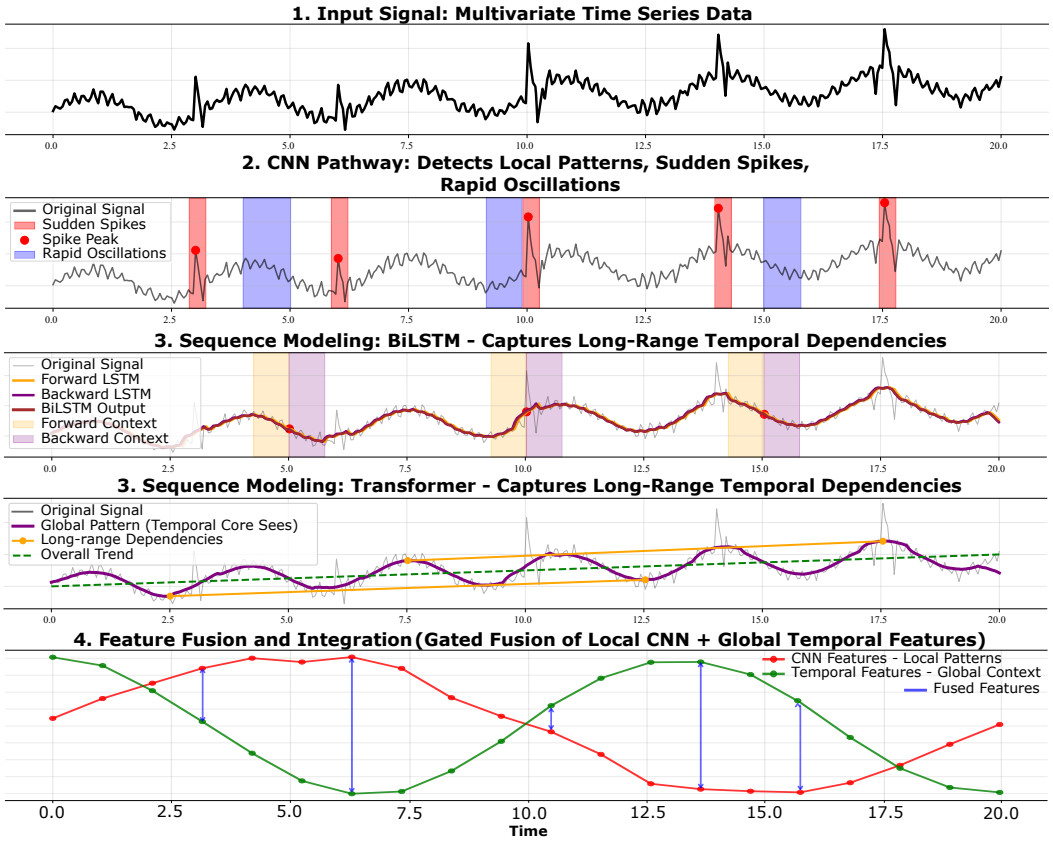

Figure 2: Multi-scale signal processing pipeline of MSTN.

the pooled representations from the two branches. The features are concatenated and passed through a SGF mechanism, which adaptively modulates the contributions of each branch. The fused representation is subsequently refined using channel-wise recalibration via SE blocks and MHA, enabling the network to emphasize the most informative feature dimensions. The final representation is normalized, regularized through dropout, and projected to the task-specific output space. For classification tasks, we employ the standard cross-entropy loss defined in Eq. 14. Figure 2 provides an overview of the end-to-end processing pipeline, from parallel feature extraction to multi-scale fusion and refinement.

3.1.2 Mathematical Modeling

We consider the following problem: Given multivariate time series samples $X_{1:L} \in \mathbb{R}^{L\times C}$ with a lookback window sequence length L and C variate features, MSTN learns temporal representations through a novel parallel architecture. We detail the complete process below.

Forward Process: The input $X \in \mathbb{R}^{B\times L\times C}$ is processed through dual complementary branches that operate simultaneously. The CNN captures local temporal patterns, while the sequence modeling (implemented as either a Transformer or a BiLSTM) captures long-range temporal dependencies. Their outputs are fused through a sophisticated gating and attention mechanism for enhanced representation learning. The input undergoes successive Conv1D operations to extract multi-scale temporal features, progressively adjusting the feature dimensionality while preserving temporal resolution. The Conv1D_k convolutional layers employ a hierarchical design in which the first layer with kernel size $k = 7$ extracts basic temporal patterns, and the second layer with kernel size $k = 5$ learns combinations of these patterns into more complex features, with padding to maintain temporal dimensions, and batch normalization stabilizes training. To aggregate these temporal features into a compact representation, we apply global average pooling across the entire sequence, where $X^\top \in \mathbb{R}^{B\times C\times L}$ denotes the channel-first representation obtained by transposing the input along the temporal and variable dimensions.

$$H^{(1)}_{\text{conv}} = \text{ReLU}(\text{BN}(\text{Conv1D}_7(X^T))) \in \mathbb{R}^{B\times 128\times L} \tag{1}$$

$$H^{(2)}_{\text{conv}} = \text{ReLU}(\text{Conv1D}_5(H^{(1)}_{\text{conv}})) \in \mathbb{R}^{B\times 64\times L} \tag{2}$$

$$\mathbf{z}_{\text{cnn}} = \text{GlobalAvgPool1D}(H^{(2)}_{\text{conv}}) \in \mathbb{R}^{B\times 64} \tag{3}$$

where each of the 64 feature channels is summarized by its average activation over time through temporal pooling. We employ two alternative sequence modeling approaches: MSTN-Transformer and MSTN-BiLSTM. First, the input sequence is projected into a 128-dimensional latent feature space using a linear embedding layer. Specifically, the projected representation is obtained as $X_p = W_pX + b_p$, where $X_p \in \mathbb{R}^{B\times L\times 128}$, and $W_p \in \mathbb{R}^{C\times 128}$ and $b_p \in \mathbb{R}^{128}$ denote the learnable weight matrix and bias of the projection layer that embed the original C-dimensional input features into a 128-dimensional latent space. In MSTN-Transformer, the architecture captures long-range dependencies through a *single* Transformer encoder composed of 4 layers with 8 attention heads. The encoder produces contextualized representations for each of the L time steps. To consolidate these temporal representations into a fixed-length global descriptor, we apply sequence mean pooling over the temporal dimension:

$$H_{\text{trans}} = \text{TransformerEnc}(X_p) \in \mathbb{R}^{B\times L\times 128}, \quad \mathbf{z}_{\text{trans}} = \frac{1}{L}\sum_{t=1}^{L} H_{\text{trans},t} \in \mathbb{R}^{B\times 128} \tag{4}$$

In MSTN-BiLSTM, bidirectional sequential dependencies are modeled by processing the projected sequence in both forward and backward directions, enabling the network to learn complex temporal dynamics and long-range contextual patterns:

$$H_{\text{bilstm}} = \text{BiLSTM}(X_p) \in \mathbb{R}^{B\times L\times 128}, \quad \mathbf{z}_{\text{bilstm}} = \frac{1}{L}\sum_{t=1}^{L} H_{\text{bilstm},t} \in \mathbb{R}^{B\times 128} \tag{5}$$

The BiLSTM consists of 2 layers with 64 hidden units per direction, capturing both causal and anti-causal temporal relationships.

The CNN branch utilizes global average pooling, and the sequence modeling branches employ sequence mean pooling. These pooling operations, along with the subsequent reshaping, collectively form the ETA module, which collapses the temporal dimension L to 1. This critical step ensures that all subsequent operations execute at $\mathcal{O}(1)$ inference cost. This design enables MSTN to achieve constant-time inference with respect to the input sequence length while preserving both local and global temporal semantics. The parallel features undergo sophisticated fusion through a multi-stage enhancement process that combines the strengths of both branches. For each MSTN variant (MSTN-Transformer or MSTN-BiLSTM):

$$\mathbf{z}_{\text{concat}} = [\mathbf{z}_{\text{cnn}}; \mathbf{z}_{\text{trans/bilstm}}] \in \mathbb{R}^{B\times 192} \tag{6}$$

where $\mathbf{z}_{\text{cnn}} \in \mathbb{R}^{B\times 64}$ and $\mathbf{z}_{\text{trans/bilstm}} \in \mathbb{R}^{B\times 128}$. The architecture then performs adaptive weighting through an SGF mechanism, illustrated in Figure 1, that learns to dynamically balance feature contributions:

$$\mathbf{z}_{\text{fused}} = \mathbf{z}_{\text{concat}} \odot \sigma(W_g \mathbf{z}_{\text{concat}} + b_g) \tag{7}$$

where for both variants: $W_g \in \mathbb{R}^{192\times 192}$, $b_g \in \mathbb{R}^{192}$. This mechanism learns to emphasize the most relevant features from each branch, creating a harmonious blend of multi-scale temporal features. The fused features are reshaped for efficient single-token processing: $\mathbf{z}_{\text{seq}} = \mathbf{z}_{\text{fused}} \otimes \mathbf{1} \in \mathbb{R}^{B\times 1\times 192}$. Channel-wise attention then enhances features through an SE block. Given the temporal dimension is already $L = 1$, the global pooling step within SE is trivially executed: $\mathbf{z}_{\text{se}} = \mathbf{z}_{\text{seq}} \odot \sigma(W_2 \text{ReLU}(W_1 \mathbf{z}_{\text{seq}}))$ where $W_1 \in \mathbb{R}^{24\times 192}$, $W_2 \in \mathbb{R}^{192\times 24}$ with reduction ratio 8. The SE mechanism amplifies informative channels while suppressing less useful ones. Feature dependencies are refined through MHA. Since the temporal dimension is condensed ($L = 1$), the MHA block functions here as a sophisticated global feature recalibration layer rather than a temporal mixer, confirming $\mathcal{O}(1)$ complexity:

$$\mathbf{z}_{\text{final}} = \text{Dropout}\left(\text{LayerNorm}\left(\text{MHA}\left(\mathbf{z}_{\text{se}}\right)\right), p = 0.3\right) \in \mathbb{R}^{B\times 192} \tag{8}$$

where 192 is the feature dimension for both MSTN-Transformer and MSTN-BiLSTM, with 4 attention heads (dim=48 per head). The refined features $\mathbf{z}_{\text{final}}$ are then passed to the final task-specific head (imputation, forecasting, or classification). The prediction head consists of a single linear layer that maps the 192-dimensional aggregated feature vector to the task-specific output shape.

Task-Specific Prediction and Loss Functions: We detail the prediction head ($\hat{Y}$ or $\hat{X}$) and the corresponding loss function $\mathcal{L}$ for each of the three tasks. The final representation $\mathbf{z}_{\text{final}}$ is passed through specialized linear heads to map the latent features to the required output space.

Time Series Imputation: The head reconstructs the input sequence $\hat{X}_{1:L} \in \mathbb{R}^{B\times L\times C}$. The objective is to minimize the masked Mean Squared Error (MSE) computed only over the observed values, where Ω is the binary mask matrix ($\Omega_{i,t,c} = 1$ if the value is observed):

$$\hat{X}_{1:L} = W_i \mathbf{z}_{\text{final}} + b_i \in \mathbb{R}^{B\times L\times C} \tag{9}$$

$$\mathcal{L}_{\text{imputation}} = \frac{1}{\|\Omega\|_1} \sum_{i=1}^{B}\sum_{t=1}^{L}\sum_{c=1}^{C} \Omega_{i,t,c} \cdot \left(\hat{X}_{i,t,c} - X_{i,t,c}\right)^2 \tag{10}$$

Long-Term Forecasting: The head produces a forecast $\hat{Y}_{1:H} \in \mathbb{R}^{B\times H\times C}$ for a prediction horizon H. The objective is to minimize the MSE across the batch, horizon, and variable dimensions:

$$\hat{Y}_{1:H} = W_f \mathbf{z}_{\text{final}} + b_f \in \mathbb{R}^{B\times H\times C} \tag{11}$$

$$\mathcal{L}_{\text{forecast}} = \frac{1}{BHC} \sum_{i=1}^{B}\sum_{t=1}^{H}\sum_{c=1}^{C} \left(\hat{Y}_{i,t,c} - Y_{i,t,c}\right)^2 \tag{12}$$

Table 1: Taxonomy of model architectures by Channel-Interaction strategy. Different state-of-the-art models align with the established Channel Interaction (CI, CD, CP) strategies. MSTN is positioned as a hybrid design (CI+CD) via early temporal aggregation to capture multi-scale dynamics. (Asym.: Asymmetry, Lag.: Lagginess, Pol.: Polarity, Gw.: group-wise, Dyn.: Dynamism, Ms.: Multi-scale)

Strategy	**Mechanism**	**Characteristic**						**Method**
		Asym.	**Lag.**	**Pol.**	**Gw.**	**Dyn.**	**Ms.**	
CI	-	-	-	-	-	-	-	PatchTST
CI	-	-	-	-	-	-	-	CycleNet
CI	-	-	-	-	-	-	-	DLinear
CI	-	-	-	-	-	-	-	Timer
CI	-	-	-	-	-	-	-	Chronos
CI	-	-	-	-	-	-	-	LLM4TS
CI	-	-	-	-	-	-	-	Time-LLM
CI	-	-	-	-	-	-	-	RevIN
CD	CNN-based	✓	-	-	-	-	-	Informer
CD	CNN-based	✓	-	-	-	-	-	Autoformer
CD	CNN-based	✓	-	-	-	-	-	FEDformer
CD	CNN-based	✓	-	-	-	-	-	TimesNet
CD	MLP-based	✓	-	-	-	-	-	TSMixer
CD	MLP-based	✓	-	-	-	-	-	TTM
CD	Transformer-based	✓	-	-	-	✓	-	iTransformer
CD	Transformer-based	✓	-	-	-	✓	-	Crossformer
CD	Transformer-based	✓	✓	-	-	-	-	VCformer
CD	Transformer-based	✓	✓	-	-	✓	-	MOIRAI
CD	Transformer-based	✓	-	-	-	✓	-	UniTS
CD	GNN-based	-	-	-	✓	-	-	GTS
CD	GNN-based	✓	-	-	-	-	✓	MSGNet
CD	GNN-based	-	✓	-	-	-	-	FourierGNN
CD	GNN-based	-	✓	-	-	-	-	FC-STGNN
CD	GNN-based	✓	-	-	-	✓	-	TPGNN
CD	Linear Channel Mixing (Factorization)	✓	-	-	-	-	-	SOFTS
CD	Others	✓	-	-	-	✓	-	C-LoRA
CP	CNN-based	✓	-	-	-	-	-	ModernTCN
CP	Transformer-based	✓	-	-	✓	-	-	DUET
CP	Transformer-based	✓	-	-	-	✓	-	MCformer
CP	Transformer-based	✓	-	-	✓	✓	-	DGCformer
CP	Transformer-based	-	-	-	-	✓	-	CM
CP	GNN-based	✓	-	-	-	-	-	MTGNN
CP	GNN-based	✓	-	✓	-	-	-	CrossGNN
CP	GNN-based	✓	-	-	✓	-	-	WaveForM
CP	GNN-based	-	-	-	-	✓	-	MTSF-DG
CP	GNN-based	✓	-	-	✓	-	-	ReMo
CP	GNN-based	✓	-	-	✓	✓	✓	Ada-MSHyper
CP	Others	✓	✓	-	-	-	-	LIFT
CP	Others	✓	-	-	✓	-	-	CCM
CI + CD	**Parallel CNN (CI) and BiLSTM/Transformer(CD) based Dual Encoder + ETA**	✓	-	-	-	✓	✓	**MSTN**

Time Series Classification: The head computes class probabilities $P \in \mathbb{R}^{B\times K}$ over K target classes. The model is optimized using the canonical Cross-Entropy (CE) Loss:

$$P = \text{Softmax}(W_c \mathbf{z}_{\text{final}} + b_c) \in \mathbb{R}^{B\times K} \tag{13}$$

$$\mathcal{L}_{\text{classify}} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{k=1}^{K} Y_{i,k}\log(P_{i,k}) \tag{14}$$

where Y is the one-hot encoded true label. This architectural modularity allows the MSTN framework to be universally applied to diverse multivariate time series tasks without structural modification.

3.2 How is MSTN different from prior works?

The established field of multivariate time–series forecasting is conventionally organized around three channel interaction strategies—CI, CD, and CP approaches Qiu et al. (2025)—as summarized in Table 1. These strategies specify how a model encodes and exploits inter-variable dependencies across the C channels of a multivariate signal.

Table 2: Comparison of architectural characteristics. Evaluation of the MSTN framework against the CI, CD, and CP Channel Interaction strategies. MSTN achieves high capacity while neutralizing the $\mathcal{O}(L^2)$ computational bottleneck.

Dimension	CI	CD	CP	(CI+CD) MSTN	Rationale
Efficiency	High	Low	Moderate	High	ETA enables $\mathcal{O}(1)$ refinement after initial encoding.
Robustness	High	Low	Moderate	High	Dual-encoder with SGF provides complementary feature stability.
Generalizability	Low	Moderate	High	High	Achieved via a single, aggregated feature vector ($\mathbf{z}_{\text{final}}$) that supports multi-task learning (Forecasting, Imputation, Classification).
Capacity	Low	High	Moderate	High	Transformer/BiLSTM path captures global dependencies.
Ease of Implementation	High	Moderate	Low	Moderate	Relies on standard components (CNN, Transformer/BiLSTM, SGF, SE, MHA) rather than complex, specialized graph or recursive structures.

CI. CI methods (e.g., DLinear, PatchTST) process each variable independently using lightweight temporal operators such as linear layers or multilayer perceptrons. Their advantage lies in linear channel cost $\mathcal{O}(C)$ and efficient temporal complexity $\mathcal{O}(L)$, though this comes at the expense of discarding cross-channel structure.

CD. CD approaches (e.g., iTransformer) explicitly model inter-variable correlation through dense attention or graph-based mechanisms. While these models offer comprehensive feature coupling, they incur high computational cost—typically $\mathcal{O}(C^2)$ or $\mathcal{O}(L^2)$—which limits their scalability and latency characteristics.

CP. CP models (e.g., MCformer, MTGNN) map the channel dimension into a low-rank latent space before applying interaction, thereby avoiding the full $\mathcal{O}(C^2)$ cost of CD models. Although channel-efficient, these models often impose simplified temporal assumptions, reducing their ability to capture multi-scale dynamics.

By combining the complementary strengths of CI and CD models with a novel temporal aggregation mechanism, MSTN delivers strong performance across forecasting, imputation, and classification tasks while maintaining the low-latency characteristics.

3.2.1 Core Innovation: Dual-Path Design and ETA

The MSTN framework integrates a dual-path temporal encoding architecture with the ETA mechanism, resulting in a principled hybrid approach that reconciles the core trade-offs among CI-, CD-, and CP-based modeling strategies as shown in Table 2. Its design is centered on two architectural innovations that jointly enhance representational fidelity and computational efficiency:

1. **Dual-branch temporal encoding.** In contrast to recent CP-oriented architectures such as MCformer Han et al. (2024b), which priorities efficiency at the expense of temporal expressiveness, MSTN explicitly preserves multi-scale temporal structure. It deploys a parallel encoding strategy in which a global sequence-modeling branch (Transformer or BiLSTM; CD strategy) captures long-range temporal dependencies, while a lightweight convolutional branch (CI strategy; $\mathcal{O}(L)$) extracts fine-grained local temporal patterns. This duality directly addresses the multi-scale nature of real-world temporal signals, offering coverage beyond what is typically afforded by efficient CP-based methods.

2. **Efficiency through ETA: collapsing $\mathcal{O}(L^2)$ to $\mathcal{O}(1)$.** MSTN applies the ETA mechanism immediately after the high-capacity encoders, collapsing the temporal dimension ($L \to 1$) via learned sequence aggregation. By performing the computationally intensive operations (e.g., $\mathcal{O}(L^2)$ Transformer self-attention) *before* this aggregation step, the subsequent refinement layers—including feature fusion, SE recalibration, MHA, and prediction modules—operate with a fixed $\mathcal{O}(1)$ cost with respect to L. This explicitly reshapes the model's complexity profile, enabling MSTN to maintain the representational power of CD architectures while achieving substantially improved inference efficiency. Unlike conventional temporal pooling, hierarchical downsampling, or token-merging strategies, ETA is not introduced as a representational shortcut, but as a principled reordering of computation. Prior approaches typically aggregate temporal information either progressively across

layers or at the final prediction stage, implicitly coupling representational capacity with inference-time complexity. In contrast, ETA explicitly allows high-capacity temporal modeling to operate on the full input sequence, while enforcing an early collapse of the temporal dimension before downstream fusion and refinement.

3.3 Baselines

This section presents the baselines used to assess MSTN on four time series tasks: (1) imputation, (2) long-term forecasting, (3) classification, and (4) generalizability study.

3.3.1 Imputation

Following the TimesNet Wu et al. (2023) protocol, we evaluate performance under the MCAR scenario with random masking ratios of $\{12.5\%, 25\%, 37.5\%, 50\%\}$. Performance is quantified using MSE and MAE. The evaluation includes recent SOTA baselines such as GPT2(3) Zhou et al. (2023), TimesNet Wu et al. (2023), PatchTST Nie et al. (2023), LightTS Zhang et al. (2022), and DLinear Zeng et al. (2022).

3.3.2 Long Term Forecasting

The proposed MSTN model is evaluated for the long-term time series forecasting task. We evaluated all methods across multiple prediction horizons $H \in \{96, 192, 336, 720\}$ for standard benchmarks and $H \in \{24, 36, 48, 60\}$ for the ILI dataset. The forecast performance is quantified using MSE and MAE. The evaluation includes the following SOTA baselines: EMTSF Alharthi et al. (2025), DARTS-TS Deng & Lindauer (2025), LLM4TS Chang et al. (2025), HiMTM Zhao et al. (2024), TIME-LLM Jin et al. (2024), MTST Zhang et al. (2024), SOFTS Han et al. (2024a), and iTransformer Liu et al. (2024).

3.3.3 Classification

The performance of classification tasks is evaluated using the standard metric of classification accuracy. The evaluation compares with recent SOTA baselines such as GPT2(6) Alharthi et al. (2025), TimesNet Wu et al. (2023), DLinear Zeng et al. (2022), ETSFormer Woo et al. (2022), FEDformer Zhou et al. (2022a), and Informer Zhou et al. (2021).

3.3.4 Generalizability Study

To evaluate generalizability between domains of the proposed MSTN framework, we benchmarked against baselines (e.g. TimesNet Wu et al. (2023), PatchTST Nie et al. (2023)) on seven publicly available international datasets spanning healthcare, activity recognition, agricultural technology, and industrial monitoring. All evaluations utilized a consistent five-fold cross-validation framework to ensure a robust and fair comparison.

3.4 Experimental Setup

3.4.1 Training and Model Configuration

The proposed MSTN model is implemented in Python 3.13.1 using PyTorch 2.7.1 and trained on an NVIDIA T400 GPU (4 GB VRAM). The architecture processes variable length sequences through the sequence modeling cores BiLSTM (128 hidden units) or Transformer (4 layers, 8 attention heads), combined with a CNN branch ($128 \rightarrow 64$ filters). Training uses AdamW Kingma & Ba (2017) optimizer with learning rate 3×10^{-4}, batch size 64, and task-specific objectives (Cross-Entropy Loss for classification, MSE for forecasting and imputation). Models train for up to 100 epochs with early stopping based on validation performance. Data preprocessing employs min-max normalization to standardize feature distributions. For forecasting and imputation benchmarks, we utilize a fixed input lookback window of $L = 96$ (except $L = 36$ for ILI) across all prediction horizons, whereas for classification tasks involving variable length sequences (e.g., JapaneseVowels), inputs are standardized to the maximum sequence length (e.g., $L = 29$) to facilitate consistent batch processing.

3.4.2 Evaluation Metrics

The performance of forecasting and imputation tasks is evaluated using two primary regression metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE). For classification tasks, we evaluate performance using Accuracy, Precision, Recall, and F1-Score to comprehensively assess predictive performance, which is particularly important for handling potential class imbalance in the datasets. Reported MSE/MAE values are averaged across all prediction horizons and all variates.

4 Experiments

This section presents a comprehensive evaluation of the proposed MSTN model. We demonstrate its performance on three time-series tasks: (1) imputation, (2) long-term forecasting, (3) classification. Additionally, we also present the generalizability study on standard time-series benchmark datasets.

4.1 Benchmark Datasets

We evaluate the proposed method across three fundamental time series tasks: forecasting, missing value imputation, and classification using established public benchmark datasets. Table 3a provides a detailed overview of all datasets. Missing value imputation is performed under a Missing Completely at Random (MCAR) setting, with masking ratios ranging from 12.5%, 25%, 37.5% and 50%. Benchmarks include ETTh1, ETTh2, ETTm1, ETTm2, Electricity, and Weather datasets Zhou et al. (2021); Trindade (2015); Köllé (2025). Imputation results are quantified using Mean Squared Error (MSE) and Mean Absolute Error (MAE) over sequence lengths of 96 time steps.

Long-term forecasting performance is assessed on nine widely-used benchmarks: ETTh1, ETTh2 (Electricity Transformer Temperature datasets of the China State Grid, recorded at an hourly resolution), ETTm1, ETTm2 (15-minute variants of the same transformer monitoring system), Electricity (ECL - hourly household consumption data from 321 U.S. clients), Traffic (hourly road occupancy rates from CalTrans loop detectors in the United States), Weather (10-minute meteorological observations from 21 U.S. stations), Exchange (daily currency exchange rates across major global economies), and ILI (weekly influenza-like illness records from the U.S. CDC). These datasets are extensively used for time-series benchmarking and are publicly available Wu et al. (2022). Forecasting horizons are set between 96, 192, 336, and 720 time steps. Long-term forecasting performance is evaluated using MSE and MAE.

The multivariate time series classification is evaluated on ten diverse datasets from the University of East Anglia (UEA) archive Bagnall et al. (2018), which includes applications from healthcare to activity recognition. To rigorously assess cross-domain transferability, we utilize seven publicly available international datasets spanning diverse applications: Human Safety/Healthcare (Fall Event Boubezoul et al. (2020), Risky Driving Rodegast et al. (2024a)); Human Activity Recognition (HAR) (UCI-HAR Reyes-Ortiz et al. (2013), PAMAP2 Reiss (2012)); Agricultural Technology/Welfare (ActBe-Calf Dissanayake et al. (2025)); and Industrial Monitoring/Predictive Maintenance (MetroPT-3 Davari et al. (2021a), NASA Turbofan Engine Saxena & Goebel (2008)).

4.2 Imputation Results

Imputation is the task of filling in missing values in a dataset. In real-world time series data, sensors can fail or data can be corrupted, which may create gaps in the data. This task tests the ability of the model to intelligently reconstruct missing data based on the surrounding context, which is vital for maintaining data quality.

As presented in Table 3b, MSTN-Transformer achieves SOTA performance in 32 out of 48 (≈66%) imputation evaluation scenarios and delivers the improved average results on 5 of the 6 benchmark datasets, while MSTN-BiLSTM demonstrates competitive performance. Following the TimesNet benchmark protocol Wu et al. (2023), we evaluate imputation performance on electricity and weather domain datasets, including ETT Zhou et al. (2021), ECL Trindade (2015), and Weather Köllé (2025). To simulate real-world scenarios, we employ random masking ratios of {12.5%, 25%, 37.5%, 50%}. This shows improved performance over

Table 3: (a) Benchmark Datasets Details. (b) Imputation Results Comparison.

(a) Benchmark Datasets for Forecasting, Imputation and Classification (10 UEA).

Task	Datasets	Features	Timesteps	Metrics	Length
Imputation	ETTh1/ETTh2	7	17,420	MSE, MAE	96
	ETTm1/ETTm2	7	69,680		
	Electricity	321	26,304		
	Weather	21	52,696		
Forecasting	ETTh1/ETTh2	7	17,420	MSE, MAE	96-720
	ETTm1/ETTm2	7	69,680		
	Electricity	321	26,304		
	Traffic	862	17,544		
	Weather	21	52,696		
	ILI	7	966		
Classification	EthanolConcentration	3	1,751	Accuracy	29-1,751
	FaceDetection	144	62		
	Handwriting	3	152		
	Heartbeat	61	405		
	JapaneseVowels	12	29		
	PEMS-SF	963	144		
	SelfRegulationSCP1	6	896		
	SelfRegulationSCP2	7	1,152		
	SpokenArabicDigits	13	93		
	UWaveGestureLibrary	3	315		

(b) Imputation Task Results Comparison. We randomly mask 12.5%, 25%, 37.5%, and 50% time points to compare the model performance under different missing degrees. **Red**/**Blue**: First/Second ranks. Tra.: Transformer, BiL.: BiLSTM, FED.: FEDformer

Models	**MSTN-Tra. (Ours)**		**MSTN BiL. (Ours)**		**GPT2(3) (2023)**		**TimesNet (2023)**		**PatchTST (2023)**		**LightTS (2022)**		**DLinear (2023)**		**FED. (2022)**		**Stationary (2022)**	
Mask R.	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1																		
12.5%	**0.024**	**0.040**	**0.032**	**0.047**	0.043	0.140	0.057	0.159	0.093	0.201	0.240	0.345	0.151	0.267	0.070	0.190	0.060	0.165
25%	**0.050**	**0.083**	**0.067**	**0.096**	0.054	0.156	0.069	0.178	0.107	0.217	0.265	0.364	0.180	0.292	0.106	0.236	0.080	0.189
37.5%	**0.083**	**0.131**	**0.103**	**0.147**	0.072	0.180	0.084	0.196	0.120	0.230	0.296	0.382	0.215	0.318	0.124	0.258	0.102	0.212
50%	0.127	**0.187**	0.148	**0.204**	**0.107**	0.216	**0.102**	0.215	0.141	0.248	0.334	0.404	0.257	0.347	0.165	0.299	0.133	0.240
Avg	**0.071**	**0.110**	0.088	**0.124**	**0.069**	0.173	0.078	0.187	0.115	0.224	0.284	0.373	0.201	0.306	0.117	0.246	0.094	0.201
ETTh2																		
12.5%	**0.009**	**0.024**	**0.022**	**0.040**	0.039	0.125	0.040	0.130	0.057	0.152	0.101	0.231	0.100	0.216	0.095	0.212	0.042	0.133
25%	**0.022**	**0.054**	**0.043**	**0.078**	0.044	0.135	0.046	0.141	0.061	0.158	0.115	0.246	0.127	0.247	0.137	0.258	0.049	0.147
37.5%	**0.038**	**0.089**	0.069	**0.121**	**0.051**	0.147	0.052	0.151	0.067	0.166	0.126	0.257	0.158	0.276	0.187	0.304	0.056	0.158
50%	0.066	**0.135**	0.102	0.170	**0.059**	**0.158**	**0.060**	0.162	0.073	0.174	0.136	0.268	0.183	0.299	0.232	0.341	0.065	0.170
Avg	**0.034**	**0.076**	0.059	**0.102**	**0.048**	0.141	0.049	0.146	0.065	0.163	0.119	0.250	0.142	0.259	0.163	0.279	0.053	0.152
ETTm1																		
12.5%	0.024	**0.038**	0.036	**0.049**	**0.017**	0.085	**0.019**	0.092	0.041	0.130	0.075	0.180	0.058	0.162	0.035	0.135	0.026	0.107
25%	0.050	**0.080**	0.072	0.099	**0.022**	**0.096**	**0.023**	0.101	0.044	0.135	0.093	0.206	0.080	0.193	0.052	0.166	0.032	0.119
37.5%	0.082	**0.124**	0.112	0.150	**0.029**	**0.111**	**0.029**	**0.111**	0.049	0.143	0.113	0.231	0.103	0.219	0.069	0.191	**0.039**	0.131
50%	0.121	0.176	0.160	0.210	**0.040**	**0.128**	**0.036**	**0.124**	0.055	0.151	0.134	0.255	0.132	0.248	0.089	0.218	0.047	0.145
Avg	0.069	**0.104**	0.095	0.127	**0.028**	**0.105**	**0.027**	0.107	0.047	0.140	0.104	0.218	0.093	0.206	0.062	0.177	0.036	0.126
ETTm2																		
12.5%	0.039	**0.048**	0.052	**0.057**	**0.017**	0.076	**0.018**	0.080	0.026	0.094	0.034	0.127	0.062	0.166	0.056	0.159	0.021	0.088
25%	0.080	0.097	0.107	0.116	**0.020**	**0.080**	**0.020**	**0.085**	0.028	0.099	0.042	0.143	0.085	0.196	0.080	0.195	**0.024**	0.096
37.5%	0.132	0.153	0.164	0.176	**0.022**	**0.087**	**0.023**	**0.091**	0.030	0.104	0.051	0.159	0.106	0.222	0.110	0.231	0.027	0.103
50%	0.187	0.215	0.225	0.241	**0.025**	**0.095**	**0.026**	**0.098**	0.034	0.110	0.059	0.174	0.131	0.247	0.156	0.276	0.030	0.108
Avg	0.109	0.128	0.137	0.147	**0.021**	**0.084**	**0.022**	**0.088**	0.029	0.102	0.046	0.151	0.096	0.208	0.101	0.215	0.026	0.099
ECL																		
12.5%	**0.002**	**0.014**	**0.034**	**0.048**	0.080	0.194	0.085	0.202	0.055	0.160	0.102	0.229	0.092	0.214	0.107	0.237	0.093	0.210
25%	**0.005**	**0.028**	0.069	**0.096**	0.087	0.203	0.089	0.206	**0.065**	0.175	0.121	0.252	0.118	0.247	0.120	0.251	0.097	0.214
37.5%	**0.008**	**0.044**	0.108	**0.147**	0.094	0.211	0.094	0.213	**0.076**	0.189	0.141	0.273	0.144	0.276	0.136	0.266	0.102	0.220
50%	**0.013**	**0.063**	0.150	**0.202**	0.101	0.220	0.100	0.221	**0.091**	0.208	0.160	0.293	0.175	0.305	0.158	0.284	0.108	0.228
Avg	**0.007**	**0.037**	0.090	**0.123**	0.090	0.207	0.092	0.210	**0.072**	0.183	0.131	0.262	0.132	0.260	0.130	0.259	0.100	0.218
Weather																		
12.5%	**0.003**	**0.016**	0.026	**0.036**	0.026	0.049	**0.025**	0.045	0.029	0.049	0.047	0.101	0.039	0.084	0.041	0.107	0.027	0.051
25%	**0.007**	**0.032**	0.051	0.070	**0.028**	**0.052**	0.029	0.052	0.031	0.053	0.052	0.111	0.048	0.103	0.064	0.163	0.029	0.056
37.5%	**0.011**	**0.049**	0.078	0.109	0.033	0.060	**0.031**	**0.057**	0.035	0.058	0.058	0.121	0.057	0.117	0.107	0.229	0.033	0.062
50%	**0.015**	0.068	0.111	0.153	0.037	0.065	**0.034**	**0.062**	0.038	**0.063**	0.065	0.133	0.066	0.134	0.183	0.312	0.037	0.068
Avg	**0.009**	**0.041**	0.066	0.092	0.031	0.056	**0.030**	**0.054**	0.033	0.056	0.055	0.117	0.052	0.110	0.099	0.203	0.032	0.059

several recent approaches, including GPT2(3) Zhou et al. (2023), TimesNet Wu et al. (2023), PatchTST Nie et al. (2023), LightTS Zhang et al. (2022), DLinear Zeng et al. (2022) and FEDformer Zhou et al. (2022b).

Figure 3 visualizes imputation results on real-world weather data with a 50% mask ratio across all evaluated methods—MSTN-Transformer, GPT2(3), TimesNet, PatchTST, LightTS, DLinear and FEDformer. The results show that MSTN-Transformer closely aligns with the ground truth temporal dynamics. Most baselines follow the general shape of the ground truth, but struggle with finer-grained temporal dynamics. In contrast, MSTN-Transformer consistently tracks both smooth trends and sharp fluctuations, indicating its effectiveness in capturing multi-scale temporal dependencies.

4.3 Long-Term Forecasting Results

Time series forecasting plays a crucial role in applications such as weather prediction, traffic management, and energy consumption planning. Long-term forecasting aims to predict future values far ahead in time. For example, it involves using historical weather data to estimate not only the temperature of tomorrow but also the conditions over the coming days or even weeks. We evaluate long-term forecasting performance using established protocols from the prior literature Nie et al. (2023); Wu et al. (2023; 2022) across nine diverse benchmark datasets. The prediction horizons are set to $H \in \{96, 192, 336, 720\}$ for the standard benchmarks and $H \in \{24, 36, 48, 60\}$ for the ILI data set. As presented in Table 4, MSTN-Transformer and MSTN-BiLSTM show improved performance compared to specialized time series forecasting architectures, including EMTSF Alharthi et al. (2025), DARTS-TS Deng & Lindauer (2025) LLM4TS Chang et al. (2025), HiMTM Zhao et al. (2024), TIME-LLM Jin et al. (2024), MTST Zhang et al. (2024), SOFTS Han et al. (2024a), and iTransformer Liu et al. (2024).

MSTN-Transformer and MSTN-BiLSTM demonstrate improved performance, ranking first and second in average MSE and MAE across multiple benchmark datasets. The variants achieve top-two positions on 6 out of 9 standard benchmarks, with MSTN-Transformer leading on 6 datasets (ETTm1, ETTh1, ECL, Traffic, Weather, ILI) and MSTN-BiLSTM securing second place while showing improved performance over other baselines. On the Traffic dataset, MSTN-Transformer achieves an average MSE of 0.019 (19.9× lower than EMTSF) while MSTN-BiLSTM delivers a competitive 0.086 MSE. Similarly, on the ECL dataset, MSTN-Transformer achieves 0.043 MSE (3.6× better than EMTSF) and MSTN-BiLSTM show robust performance. Both MSTN variants dominate the ILI dataset with MSEs of 0.165 and 0.166, outperforming TIME-LLM (a large-language model) by 8.7×. Regarding the Exchange dataset, due to the absence of reported results in the baseline literature for the comparable models, we present MSTN performance independently. MSTN-BiLSTM excels on the Exchange dataset, achieving an average MSE of 0.408 and MAE of 0.526. In comparison, MSTN-Transformer scores 0.668 MSE and 0.675 MAE on the same dataset. This demonstrate the architectural flexibility of the MSTN framework. This consistent SOTA performance across diverse domains validates the effectiveness of multi-scale temporal modeling in both Transformer and BiLSTM configurations.

Figure 4 illustrates the 336-step forecasting performance of MSTN-Transformer, SOFTS, and iTransformer under the input-96-predict-336 setting. MSTN-Transformer achieves the lowest MSE and MAE and exhibits the realistic forecasting behavior, showing smoother trajectories with fewer abrupt fluctuations. Importantly, MSTN-Transformer consistently demonstrates better phase alignment with the ground truth, with predicted peaks and troughs occurring at nearly identical temporal locations, accurately preserving the periodic structure. Such temporal fidelity is critical for practical forecasting scenarios, where correct timing of events is often more important than exact amplitude matching. Although MSTN-Transformer produces slightly reduced peak amplitudes, this conservative behavior reflects an error-minimizing bias of MSE/MAE-based optimization, favoring stable trend estimation over aggressive extrapolation. In particular, in the late horizon, MSTN-Transformer adapts to the upward distribution shift of the ground truth and tracks the rising structural trend, whereas EMTSF, SOFTS, and iTransformer remain largely stationary and exhibit more erratic variations. Overall, these results demonstrate the effective long-horizon forecasting capability, robustness, and non-stationary adaptation ability of MSTN-Transformer. For clarity and readability, we visualize only three representative and competitive baselines, while quantitative results for all compared models are reported in Table 4.

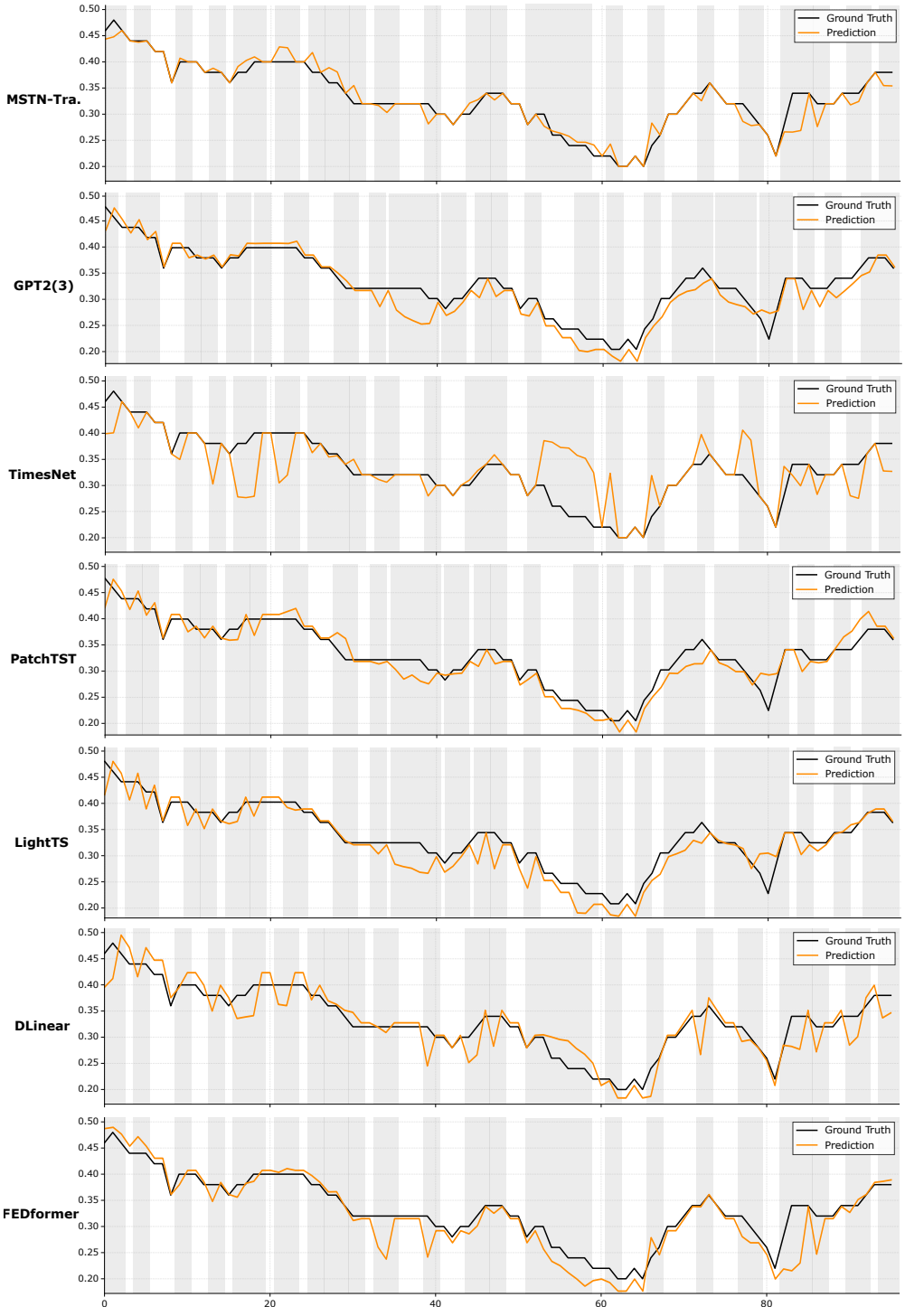

Figure 3: Visualization of weather imputation results under 50% mask ratio. The black lines represent ground truth and the orange lines represent predicted values.

4.4 Time-Series Classification Results

Classification involves categorizing entire time series sequences. It is used to identify which type of event or pattern the sequence represents. For example, this could be the classification of a segment of sensor data as

Table 4: Multivariate long-term forecasting results. Input lookback window $L = 96$ ($L = 36$ for ILI). Prediction horizons $H \in \{96, 192, 336, 720\}$ ($H \in \{24, 36, 48, 60\}$ for ILI). Avg is averaged from all four prediction lengths. **Red**/**Blue**: First/Second ranks. iTransf.: iTransformer

Models	MSTN-Tra. Ours		MSTN BiL. Ours		EMTSF 2025		DARTS-TS 2025		LLM4TS 2025		HiMTM 2025		TIME-LLM 2024		MTST 2024		SOFTS 2024		iTransf. 2024	
H-Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1																				
96	**0.052**	**0.174**	**0.168**	0.334	0.271	**0.325**	0.283	0.330	0.285	0.343	0.280	0.331	0.272	0.334	0.286	0.338	0.325	0.361	0.334	0.368
192	**0.103**	**0.242**	**0.111**	**0.258**	0.322	0.351	0.320	0.354	0.324	0.366	0.321	0.357	0.310	0.358	0.327	0.366	0.375	0.389	0.377	0.391
336	**0.127**	**0.271**	**0.197**	**0.350**	0.350	0.370	0.357	0.377	0.353	0.385	0.347	0.378	0.352	0.384	0.362	0.389	0.405	0.412	0.426	0.420
720	**0.151**	**0.301**	**0.166**	**0.314**	0.414	0.404	0.418	0.412	0.408	0.419	0.395	0.411	0.383	0.411	0.414	0.421	0.466	0.447	0.491	0.459
Avg	**0.123**	**0.262**	**0.161**	**0.314**	0.339	0.362	0.345	0.368	0.342	0.378	0.336	0.369	0.329	0.372	0.347	0.378	0.393	0.403	0.407	0.410
ETTm2																				
96	0.201	0.325	0.238	0.390	**0.156**	**0.240**	0.162	**0.244**	0.165	0.254	0.164	0.254	**0.161**	0.253	0.162	0.251	0.180	0.261	0.180	0.264
192	0.373	0.467	0.426	0.514	**0.212**	**0.280**	0.223	**0.285**	0.220	0.292	0.221	0.291	**0.219**	0.293	0.220	0.291	0.246	0.306	0.250	0.309
336	0.345	0.477	0.517	0.594	**0.263**	**0.315**	0.274	**0.320**	**0.268**	0.326	0.273	0.326	0.271	0.329	0.272	0.326	0.319	0.352	0.311	0.348
720	0.411	0.509	0.576	0.614	**0.351**	**0.371**	0.354	**0.373**	**0.350**	0.380	0.355	0.378	0.352	0.379	0.358	0.379	0.405	0.401	0.412	0.407
Avg	0.333	0.445	0.439	0.528	**0.245**	**0.301**	0.253	**0.305**	**0.251**	0.313	0.253	0.312	**0.251**	0.313	0.253	0.312	0.287	0.330	0.288	0.332
ETTh1																				
96	**0.142**	**0.285**	**0.298**	0.459	0.359	**0.384**	0.365	0.385	0.371	0.394	0.355	0.386	0.362	0.392	0.358	0.390	0.381	0.399	0.386	0.405
192	**0.144**	**0.293**	**0.288**	0.446	0.399	**0.411**	0.405	0.411	0.403	0.412	0.401	0.417	0.398	0.418	0.396	0.414	0.435	0.431	0.441	0.436
336	**0.152**	**0.307**	**0.163**	**0.314**	0.418	0.422	0.437	0.433	0.420	0.422	0.420	0.429	0.430	0.427	0.391	0.420	0.480	0.452	0.487	0.458
720	**0.181**	**0.349**	**0.223**	**0.383**	0.436	0.454	0.448	0.462	0.422	0.444	0.425	0.447	0.442	0.457	0.430	0.457	0.499	0.488	0.503	0.491
Avg	**0.155**	**0.309**	**0.243**	**0.401**	0.403	0.417	0.414	0.423	0.404	0.418	0.400	0.419	0.408	0.423	0.394	0.420	0.449	0.442	0.454	0.447
ETTh2																				
96	0.297	0.429	0.384	0.493	**0.262**	**0.324**	0.276	0.332	0.269	0.332	0.273	0.334	0.268	0.328	**0.257**	**0.326**	0.297	0.347	0.297	0.349
192	**0.300**	0.436	0.401	0.505	0.328	**0.371**	0.344	0.377	0.328	0.377	0.334	0.371	0.329	0.375	**0.309**	**0.361**	0.373	0.394	0.380	0.400
336	0.377	0.495	0.383	0.501	**0.347**	**0.387**	0.377	0.406	0.353	0.396	0.353	0.398	0.368	0.409	**0.302**	**0.366**	0.410	0.426	0.428	0.432
720	0.400	0.509	0.445	0.530	**0.372**	0.420	0.406	0.435	0.383	0.425	**0.371**	**0.412**	0.381	0.417	**0.372**	**0.416**	0.411	0.433	0.427	0.445
Avg	0.349	0.472	0.403	0.507	**0.329**	**0.374**	0.351	0.387	0.333	0.382	0.332	0.379	0.334	0.383	**0.310**	**0.367**	0.373	0.400	0.383	0.407
ECL																				
96	**0.041**	**0.161**	0.558	0.585	**0.126**	**0.217**	0.129	0.217	0.128	0.223	0.129	0.220	0.131	0.224	0.127	0.222	0.143	0.233	0.148	0.240
192	**0.042**	**0.163**	0.372	0.464	**0.144**	**0.234**	0.147	0.234	0.146	0.240	0.147	0.238	0.152	0.241	**0.144**	0.238	0.158	0.248	0.162	0.253
336	**0.042**	**0.162**	0.382	0.471	0.158	**0.248**	0.164	0.253	0.163	0.258	**0.157**	0.249	0.160	0.248	0.162	0.256	0.178	0.269	0.178	0.269
720	**0.047**	**0.173**	0.469	0.521	**0.190**	**0.277**	0.186	0.275	0.200	0.292	0.198	0.285	0.192	0.298	0.199	0.289	0.218	0.305	0.225	0.317
Avg	**0.043**	**0.165**	0.445	0.510	**0.154**	**0.244**	0.157	0.245	0.159	0.253	0.157	0.248	0.158	0.252	0.158	0.251	0.174	0.264	0.178	0.270
Traffic																				
96	**0.017**	**0.110**	**0.085**	**0.202**	0.343	0.225	0.358	0.240	0.372	0.259	0.358	0.240	0.362	0.248	0.356	0.244	0.376	0.251	0.395	0.268
192	**0.018**	**0.111**	**0.086**	**0.205**	0.369	0.238	0.386	0.256	0.391	0.265	0.368	0.248	0.374	0.247	0.375	0.251	0.398	0.261	0.417	0.276
336	**0.020**	**0.112**	**0.086**	**0.204**	0.382	0.242	0.398	0.262	0.405	0.275	0.379	0.250	0.385	0.271	0.386	0.256	0.415	0.269	0.433	0.283
720	**0.020**	**0.113**	**0.087**	**0.207**	0.424	0.270	0.434	0.284	0.437	0.292	0.430	0.276	0.430	0.288	0.425	0.279	0.447	0.287	0.467	0.302
Avg	**0.019**	**0.111**	**0.086**	**0.205**	0.379	0.243	0.394	0.260	0.401	0.273	0.384	0.254	0.388	0.264	0.386	0.257	0.409	0.267	0.428	0.282
Weather																				
96	**0.109**	0.263	**0.110**	0.264	0.138	**0.177**	0.148	**0.194**	0.147	0.196	0.141	0.182	0.147	0.201	0.150	0.199	0.166	0.208	0.174	0.214
192	**0.111**	0.264	**0.111**	0.265	0.181	**0.220**	0.195	**0.239**	0.191	0.238	0.188	0.228	0.189	0.234	0.194	0.240	0.217	0.253	0.221	0.254
336	**0.114**	**0.268**	**0.115**	0.269	0.230	**0.260**	0.252	0.282	0.241	0.277	0.240	0.273	0.262	0.279	0.246	0.281	0.282	0.300	0.278	0.296
720	**0.128**	**0.286**	**0.132**	**0.289**	0.304	0.315	0.323	0.331	0.313	0.329	0.312	0.322	0.304	0.316	0.319	0.333	0.356	0.351	0.358	0.347
Avg	**0.115**	0.270	**0.117**	0.272	0.213	**0.243**	0.229	**0.262**	0.223	0.260	0.220	0.251	0.225	0.257	0.227	0.263	0.255	0.278	0.258	0.278
Exchange																				
96	0.495	0.596	**0.231**	**0.401**	-	-	**0.088**	**0.210**	-	-	-	-	-	-	-	-	-	-	-	-
192	0.456	0.566	**0.278**	**0.437**	-	-	**0.186**	**0.308**	-	-	-	-	-	-	-	-	-	-	-	-
336	0.906	0.803	**0.303**	**0.469**	-	-	**0.350**	**0.428**	-	-	-	-	-	-	-	-	-	-	-	-
720	**0.815**	**0.734**	**0.818**	0.795	-	-	0.888	**0.689**	-	-	-	-	-	-	-	-	-	-	-	-
Avg	0.668	0.675	**0.408**	**0.526**	-	-	**0.378**	**0.409**	-	-	-	-	-	-	-	-	-	-	-	-
ILI																				
24	**0.129**	**0.254**	**0.161**	**0.328**	1.617	0.732	-	-	-	-	-	-	1.285	0.727	-	-	-	-	-	-
36	**0.155**	**0.287**	**0.167**	**0.333**	1.586	0.728	-	-	-	-	-	-	1.404	0.814	-	-	-	-	-	-
48	**0.180**	**0.323**	**0.168**	**0.334**	1.587	0.753	-	-	-	-	-	-	1.523	0.807	-	-	-	-	-	-
60	**0.190**	**0.332**	**0.169**	**0.334**	1.560	0.768	-	-	-	-	-	-	1.531	0.854	-	-	-	-	-	-
Avg	**0.165**	**0.302**	**0.166**	**0.332**	1.588	0.745	-	-	-	-	-	-	1.436	0.801	-	-	-	-	-	-

walking, running, falling, or collision activity, which is fundamental for automated monitoring and diagnosis systems. Performance is evaluated using the standard metric of classification accuracy.

Table 5a presents the classification results across 10 UEA standard benchmark datasets. The proposed MSTN framework is compared against SOTA time-series classification methods, including TimesNet Wu et al. (2023), GPT-2(6) Zhou et al. (2023), FEDformer Zhou et al. (2022a), and LightTS Zhang et al. (2022). MSTN demonstrates competitive performance in multiple datasets. MSTN-Transformer and MSTN-

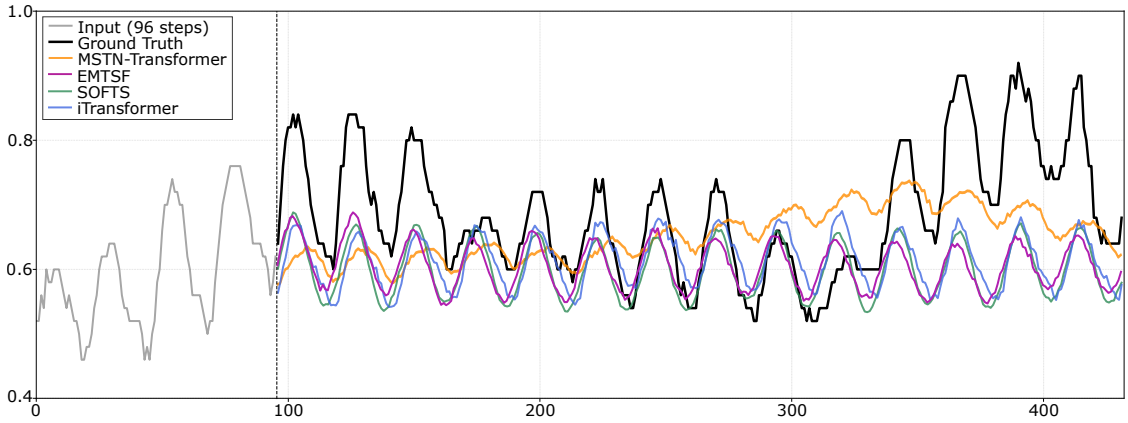

Figure 4: Visualization of weather predictions by different models under the input-96-predict-336 setting.

Table 5: Comprehensive evaluation of MSTN across diverse time-series benchmarks.

(a) Classification accuracy (%) comparison between MSTN (MSTN- Transformer and MSTN-BiLSTM) and SOTA baselines. **Red**/**Blue**: First/Second accuracy ranks.

Dataset	MSTN Trans. (ours)	MSTN BiL. (ours)	GPT2 (6) (2023)	Times-Net (2023)	Light TS (2022)	DLinear (2023)	Flow. (2022)	ETS. (2022)	FED. (2022a)	Stat. (2022)	Auto. (2021)	Py. (2021a)	In. (2021)	Re. (2020)	Trans. (2017)
Ethanol	33.92	**44.76**	31.9	**35.7**	29.7	32.6	33.8	28.1	31.2	32.7	31.6	30.8	31.6	31.9	32.7
FaceDetect	65.37	55.58	67.3	**68.6**	67.5	68.0	67.6	66.3	66.0	68.0	**68.4**	65.7	67.0	**68.6**	67.3
Handwriting	**58.50**	**89.00**	32.0	32.1	26.1	27.0	33.8	32.5	28.0	31.6	36.7	29.4	32.8	27.4	32.0
Heartbeat	**100**	**100**	76.1	78.0	75.1	75.1	77.6	71.2	73.7	73.7	74.6	75.6	**80.5**	77.1	76.1
JapaneseV	**99.29**	**100**	98.6	98.4	96.2	96.2	98.9	95.9	98.4	**99.2**	96.2	98.4	98.9	97.8	98.7
PEMS-SF	**93.18**	**92.12**	82.1	89.6	88.4	75.1	83.8	86.0	80.9	87.3	82.7	83.2	81.5	82.7	82.1
SCP1	**93.32**	91.15	**93.2**	91.8	89.8	87.3	92.5	89.6	88.7	89.4	84.0	88.1	90.1	90.4	**92.2**
SCP2	56.63	56.58	**59.4**	**57.2**	51.1	50.5	56.1	55.0	54.4	**57.2**	50.6	53.3	53.3	56.7	53.9
SpokenArabic	98.47	97.93	99.2	99.0	**100**	81.4	98.8	**100**	**100**	**100**	**100**	**99.6**	**100**	97.0	98.4
UWave	**89.77**	**97.73**	88.1	85.3	80.3	82.1	86.6	85.0	85.3	87.5	85.9	83.4	85.6	85.6	85.6
Avg	**78.85**	**82.48**	72.79	73.67	70.43	65.53	72.85	70.96	70.66	72.73	71.07	70.75	72.13	71.52	71.90

(b) Generalizability Study: Performance comparison of MSTN-BiLSTM and MSTN-Transformer with TimesNet and PatchTST across seven international cross-domain datasets. **Red**/**Blue**: First/Second accuracy ranks; **Cyan**/**Violet**: First/Second inference time ranks.

				MSTN							TimesNet		PatchTST		
Dataset	Det.	Dom.	Cnt.	A.Block	Acc.(%)	F1	Prec.	Rec.	Size	I.Time	Acc.(%)	I.Time	Acc.(%)	I.Time	Prior Acc.
Rodegast Rodegast et al. (2024a)	Sim.	Hum.	DE.	BiLSTM	**99.20**	0.992	0.993	0.991	4.39	**0.81**	99.18	5.07	98.64	4.80	91.00[RF,GB]Rodegast et al. (2024b)
				Transf.	**99.53**	0.947	0.951	0.942	0.54	0.155					
BoubezoulBoubezoul et al. (2020)	Real.	Hum.	FR.	BiLSTM	**93.75**	0.945	0.940	0.950	4.16	**2.79**	**93.00**	4.20	91.37	4.68	91.59[DT]Elwy et al. (2023)
				Transf.	91.76	0.927	0.921	0.934	0.96	0.24					
UCI-HARReyes-Ortiz et al. (2013)	Act.	Hum.	IT.	BiLSTM	**96.59**	0.965	0.966	0.965	7.14	2.03	91.38	45.30	93.21	4.36	83.35[Km,NB]Ismi et al. (2016)
				Transf.	**95.58**	0.956	0.958	0.955	29.98	**3.48**					
PAMAP2Reiss (2012)	Phy.	Hum.	US.	BiLSTM	**99.69**	0.996	0.996	0.996	0.76	0.086	95.13	0.46	98.02	3.21	90.00[kNN]Reiss & Stricker (2012)
				Transf.	**99.52**	0.995	0.995	0.995	1.58	**0.20**					
ActBeC.Dissanayake et al. (2025)	Calf.	Anim.	IE.	BiLSTM	**93.00**	0.923	0.923	0.923	4.14	**1.76**	**90.65**	8.53	62.45	2.20	84.00[RCCV]Dissanayake et al. (2025)
				Transf.	88.14	0.883	0.873	0.913	0.89	0.14					
MetroPT3Davari et al. (2021a)	Met.	Mech.	PT.	BiLSTM	**93.10**	0.930	0.930	0.931	0.57	0.042	93.00	0.09	81.67	0.19	62.00[SAE]Davari et al. (2021b)
				Transf.	**93.83**	0.938	0.940	0.938	1.30	**0.19**					
NASASaxena & Goebel (2008)	Eng.	Mech.	US.	BiLSTM	20.24†	–	–	–	4.25	**1.48**	**15.56**†	4.52	31.59†	2.31	–
				Transf.	**11.26**†	–	–	–	1.31	0.45					

Abbreviations: Det: Details (Sim: Simulator PTW data, Real: Real PTW Data, Act: Activity human, Phy: Physical activity, Calf: Calf Behavior, Met: Metro predictive maintenance, Eng: Engine sensor), Dom: Domain (Hum: Human Safety, Anim: Animal welfare, Mech: Mechanical), Cnt: Country (IN: India, FR: France, DE: Germany, IT: Italy, US: USA, IE: Ireland, PT: Portugal), A.Block: Architecture block (Transf. Transformer), I.Time: Inference time, †RMSE values.

BiLSTM achieves perfect classification (100%) on Heartbeat and near-perfect performance on JapaneseV (99.29%), while MSTN-BiLSTM achieves improved accuracy on Ethanol (44.76%), Handwriting (89.00%), JapaneseVowels (100%) and PEMS-SF (92.12%). In terms of overall average accuracy, MSTN-BiLSTM ranks first (82.48%) and MSTN-Transformer ranks second (78.85%) among all compared methods, confirming the consistent superiority of the proposed MSTN framework.

4.5 Generalizability Study

A robust model should not be a specialist in just one area; it should adapt to new tasks and data distributions across different data and domains. This experiment evaluates the ability of the MSTN to transfer its knowledge across diverse domains, from human activity recognition to mechanical fault prediction. As summarized in Table 5b, the proposed MSTN framework demonstrates generalization in seven international benchmark data sets such as human safety, activity recognition, animal welfare, and mechanical prognostics. Without domain-specific tuning, MSTN shows improved performance over the latest time series SOTA models, including TimesNet Wu et al. (2023) and PatchTST Nie et al. (2023).

MSTN achieves competitive performance across fundamentally different domains, demonstrating its capabilities as a universal temporal model. In human safety applications, MSTN-Transformer reaches 99.53% accuracy on German collision prediction data Rodegast et al. (2024a), substantially outperforming prior work (91.0% RF,GB Rodegast et al. (2024b)). For activity recognition, MSTN-BiLSTM achieves 96.59% accuracy on the UCI-HAR dataset Reyes-Ortiz et al. (2013), representing a 13.2% improvement over TimesNet. The framework also excels in real-world fall detection with 93.75% accuracy, surpassing dedicated DT-based methods. MSTN-BiLSTM also shows strong performance in animal welfare evaluation Dissanayake et al. (2025), achieving 93.00% accuracy on calf behavior recognition. Furthermore, MSTN-Transformer demonstrates better capability in mechanical prognostics, achieving 11.26 RMSE on the NASA Turbofan data set Saxena & Goebel (2008) and significantly outperforming all baseline predictors in the estimate of the remaining useful life.

Beyond accuracy, Table 5b highlights the structural impact of our ETA mechanism. While traditional models maintain full sequence complexity throughout deep layers, MSTN restricts this heavy computation to the initial encoding stage. Consequently, MSTN-Transformer achieves an inference time of 0.155 ms on the Rodegast dataset approximately 32× faster than TimesNet (5.07 ms) and 30× faster than PatchTST (4.80 ms). This confirms that MSTN effectively breaks the traditional trade-off between modeling capacity and latency.

4.6 Ablation Study

An ablation study systematically removes individual components from the full model to evaluate their contributions. Table 6 presents the results for MSTN-Transformer across six diverse benchmarks ECL (electricity), Weather, ETTm1, ETTh2, Traffic, ILI and for MSTN-BiLSTM on three datasets: Weather, ETTm2, and Traffic. For MSTN-Transformer, five variants are tested: w/o CNN, w/o Transformer, w/o SE, w/o MHA, and w/o SGF. While in MSTN-BiLSTM architecture, w/o BiLSTM denotes the variant of MSTN-BiLSTM without the BiLSTM layer.

In the MSTN-Transformer variant, the results consistently show that the full MSTN-Transformer model achieves the best performance (ranked first in MSE/MAE) across nearly all prediction horizons and datasets. Removing the core Transformer module (w/o Transformer) leads to the most severe performance drop in most cases. For instance, on ETTh2 (a challenging dataset with high volatility), the MSE increases substantially from 0.297 (full model) to 0.385 at horizon 96 when the Transformer is ablated, highlighting its critical role in modeling complex temporal patterns. Similarly, the CNN branch (w/o CNN) is highly important, especially on datasets like Traffic and ECL where local features are crucial; its removal causes noticeable degradation (e.g., Traffic H=96 MSE rises from 0.017 to 0.020). The SE blocks (w/o SE) also contribute significantly, often yielding the second-best or competitive results when present. For example, on Weather, the SE-ablated variant remains close to the full model, but on ETTm1, its removal increases MSE from 0.127 to 0.140 at H=336. The MHA (w/o MHA) and self-gated fusion (w/o SGF) modules further enhance robustness, with their absence leading to consistent, though sometimes smaller, declines across datasets.

For MSTN-BiLSTM, the full model outperforms all ablated variants. Removing the BiLSTM layer (w/o BiLSTM) results in significant deterioration, particularly on complex datasets like ETTm2 where MSE at H=720 jumps from 0.576 to 0.671. This underscores the importance of recurrent modeling in capturing sequential dependencies. In summary, the ablation study validates that each architectural component is essential. The Transformer or BiLSTM, convolutional branch, and attention-based mechanisms operate

Table 6: Ablation study of MSTN-Transformer and MSTN-BiLSTM. $H \in \{96, 192, 336, 720\}$ and for ILI $H \in \{24, 36, 48, 60\}$. **Red**/**Blue**: First/Second ranks.

Methods	MSTN Full		w/o CNN		w/o Transf.		w/o SE		w/o MHA		w/o SGF	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
MSTN Transf.												
ECL												
96	**0.041**	**0.161**	0.045	0.165	0.048	0.173	**0.042**	**0.163**	0.043	0.164	0.045	0.168
192	**0.042**	**0.163**	0.046	0.169	0.048	0.173	0.045	**0.166**	**0.044**	**0.166**	0.047	0.170
336	**0.042**	**0.162**	0.046	0.168	0.050	0.178	0.046	**0.167**	**0.045**	0.169	0.049	0.168
720	**0.047**	**0.173**	0.051	0.179	0.055	0.186	**0.049**	**0.175**	0.050	0.178	0.053	0.179
Weather												
96	**0.109**	**0.263**	0.112	0.266	0.118	0.273	**0.110**	**0.264**	0.111	0.265	0.112	0.265
192	**0.111**	**0.264**	0.115	0.269	0.120	0.275	**0.112**	**0.265**	0.113	**0.265**	0.112	0.265
336	**0.114**	**0.268**	0.120	0.274	0.123	0.279	**0.115**	**0.269**	0.116	0.270	0.116	0.270
720	**0.128**	**0.286**	0.138	0.296	0.136	0.294	**0.130**	**0.287**	0.131	0.289	**0.130**	**0.287**
ETTm1												
96	**0.052**	**0.174**	0.062	0.186	0.089	0.227	0.061	0.185	0.057	0.178	**0.055**	**0.177**
192	**0.104**	**0.242**	0.106	0.244	0.105	0.243	0.109	0.247	**0.103**	**0.244**	0.105	0.245
336	**0.127**	**0.271**	0.146	0.289	0.156	0.307	**0.140**	**0.283**	0.151	0.296	0.147	0.290
720	**0.151**	**0.301**	0.155	0.306	0.172	0.319	**0.142**	**0.288**	0.159	0.308	0.158	0.307
ETTh2												
96	**0.297**	**0.429**	0.370	0.492	0.385	0.498	**0.374**	**0.482**	0.377	0.491	0.390	0.485
192	**0.300**	**0.436**	0.404	0.511	0.391	0.505	0.392	0.506	**0.382**	**0.496**	0.395	0.501
336	**0.377**	**0.495**	0.415	0.520	0.410	0.518	0.396	0.508	**0.387**	**0.499**	0.396	0.505
720	**0.400**	**0.509**	0.428	0.524	0.430	0.525	**0.426**	**0.523**	0.426	0.531	0.439	0.531
Traffic												
96	**0.017**	**0.110**	0.020	0.113	0.026	0.128	**0.019**	**0.112**	0.021	0.116	0.020	**0.112**
192	**0.018**	**0.111**	0.020	0.115	0.026	0.127	**0.020**	**0.113**	0.023	0.116	**0.020**	**0.113**
336	**0.020**	**0.112**	0.022	0.118	0.029	0.135	**0.021**	0.115	0.024	0.118	**0.021**	**0.114**
720	**0.020**	**0.113**	0.026	0.128	0.026	0.128	**0.022**	**0.118**	0.025	0.119	0.025	0.120
ILI												
24	**0.129**	**0.254**	0.140	0.272	0.139	0.274	**0.131**	**0.255**	**0.131**	**0.255**	0.133	0.256
36	**0.155**	**0.287**	0.167	0.301	0.168	0.307	0.161	0.298	**0.157**	**0.293**	0.168	0.304
48	**0.180**	**0.323**	0.208	0.344	0.193	0.332	**0.182**	**0.324**	0.183	0.325	0.191	0.328
60	**0.190**	**0.332**	0.197	0.338	0.199	0.339	**0.193**	**0.333**	0.194	0.334	0.194	0.338

MSTN BiLSTM

Methods	MSTN Full		w/o CNN		w/o BiLSTM		w/o SE		w/o MHA		w/o SGF	
Weather												
96	**0.110**	**0.264**	0.112	0.266	0.119	0.274	**0.111**	**0.264**	**0.111**	**0.265**	0.112	0.266
192	**0.111**	**0.265**	0.113	0.267	0.120	0.275	**0.112**	**0.266**	0.113	0.267	0.113	**0.266**
336	**0.115**	**0.269**	0.116	0.270	0.123	0.279	**0.116**	**0.270**	0.117	0.271	0.118	0.272
720	**0.132**	**0.289**	0.138	0.295	0.139	0.296	**0.134**	**0.294**	0.135	**0.294**	**0.134**	0.298
ETTm2												
96	**0.238**	**0.390**	0.243	0.400	0.264	0.430	0.262	0.394	**0.240**	**0.392**	0.261	0.399
192	**0.426**	**0.514**	0.501	0.560	0.528	0.617	**0.408**	**0.510**	0.419	0.535	0.411	0.512
336	**0.517**	**0.594**	0.581	0.623	0.571	0.612	**0.535**	**0.592**	0.536	0.593	0.626	0.655
720	**0.576**	**0.614**	0.660	0.651	0.671	0.665	**0.638**	**0.645**	0.829	0.738	0.641	0.647
Traffic												
96	**0.085**	**0.202**	0.089	0.208	0.090	0.209	**0.086**	**0.203**	0.087	0.204	0.087	0.205
192	**0.087**	**0.205**	0.092	0.218	0.101	0.221	**0.088**	**0.207**	**0.088**	0.208	**0.088**	0.209
336	**0.086**	**0.204**	0.095	0.224	0.105	0.227	**0.089**	**0.213**	0.090	0.214	0.090	0.215
720	**0.087**	**0.207**	0.098	0.227	0.107	0.229	**0.091**	0.218	**0.091**	**0.217**	0.093	0.218

synergistically, with the full integrated model delivering the best overall forecasting accuracy across diverse temporal datasets.

4.7 Model Complexity and Edge-AI Deployability

4.7.1 Complexity Analysis

In this analysis, we denote the length of the lookback window as L, the number of variate features (channels) as C, and the task-specific output dimension as K. We use C for channels to align with notation in common literature such as SOFTS Han et al. (2024a), where K is often referred to as the forecasting horizon (H). We simplify the asymptotic complexity expressions by omitting the model dimension d, assuming $d \approx C$.

The MSTN framework addresses the fundamental trade-off in temporal modeling between representational capacity and computational efficiency. Although the proposed variants exhibit distinct encoding complexities, MSTN-BiLSTM maintains strict linearity with respect to sequence length and MSTN-Transformer incurring a quadratic cost. They both converge on a unified efficiency strategy: ETA. This is achieved because both the parallel CNN and sequence modeling branches utilize immediate temporal pooling to eliminate sequence-length dependence in subsequent layers. Specifically, both the parallel CNN and sequence modeling branches (Transformer/BiLSTM) independently condense their temporal features via Global Average Pooling and Sequence Mean Pooling, respectively (reducing the sequence dimension from L to 1). Following concatenation, the subsequent SGF, SE-Block, and MHA refinement stages operate on static feature vectors. This strategic design ensures that the refinement pipeline scales as $\mathcal{O}(C^2)$ with respect to channels but remains constant-time ($\mathcal{O}(1)$) with respect to the input length L, effectively isolating the heavy temporal computation to the initial stage.

Component-Wise Analysis of MSTN-Transformer: The complexity is dominated by the sequence encoding stage. First, the Multi-Scale CNN extracts local features with linear complexity $\mathcal{O}(CL)$. Second, the Transformer Encoder applies self-attention; as this requires computing an $L \times L$ attention matrix, it introduces the dominant quadratic term $\mathcal{O}(CL^2)$. Following this, Temporal Pooling aggregates the sequence with linear cost $\mathcal{O}(CL)$. Crucially, the Refinement stage operates on the pooled vector, scaling with channel interactions as $\mathcal{O}(C^2)$. Finally, the Projection Head maps features to the output with complexity $\mathcal{O}(CK)$. Summing these components, the total asymptotic complexity is $\mathcal{O}(CL^2+C^2+CK)$, which scales quadratically with sequence length L.

Component-Wise Analysis of MSTN-BiLSTM: In contrast, the MSTN-BiLSTM variant achieves strict linear efficiency with respect to sequence length. The BiLSTM Encoder processes the sequence recurrently without generating a global attention map, resulting in a linear complexity of $\mathcal{O}(CL)$. Similar to the Transformer variant, Temporal Pooling ($\mathcal{O}(CL)$) condenses the time dimension. The Refinement stage contributes $\mathcal{O}(C^2)$, and the Projection Head contributes $\mathcal{O}(CK)$. Since the encoder avoids quadratic operations in L, the total start-to-end complexity remains highly efficient at $\mathcal{O}(CL + C^2 + CK)$, which scales linearly with L.

This theoretical efficiency is empirically validated by the structural footprint of the model. As verified through our implementation, the MSTN-Transformer variant maintains a fixed core of ∼1.04 million parameters, while the MSTN-BiLSTM variant is highly optimized with only ∼0.40 million parameters. By ensuring a task-agnostic backbone, MSTN maintains consistent complexity across diverse datasets, meeting the requirement for architectural invariance. The transition from $\mathcal{O}(L)$ or $\mathcal{O}(L^2)$ in the encoding stage to $\mathcal{O}(1)$ in the refinement stage is the main reason for this low parameter count, allowing the model to perform effective feature recalibration without an additive parameter penalty.

4.7.2 Comparative Analysis and Edge-AI Deployability

Recent efficient architectures demonstrate varying complexity profiles: linear models such as DLinear Zeng et al. (2022) achieve $\mathcal{O}(CL)$ complexity, while spectral approaches like FEDformer Zhou et al. (2022a) and TimesNet Wu et al. (2023) achieve $\mathcal{O}(CL \log L)$ complexity. Following the notation in SOFTS Han et al. (2024a), their model achieves $\mathcal{O}(CL + CH)$ complexity via stochastic pooling, while iTransformer ($\mathcal{O}(C^2 + CL + CH)$), PatchTST ($\mathcal{O}(CL^2 + CH)$), and Standard Transformers ($\mathcal{O}(N \cdot CL^2)$) incur heavier costs.

Against this landscape, the proposed MSTN-BiLSTM achieves strict linear temporal complexity $\mathcal{O}(CL+C^2+CK)$. Theoretically, this matches the optimal efficiency of SOFTS in the predominant real-world scenario where sequence length exceeds channel dimension ($L \gg C$). While SOFTS retains a theoretical advantage in high-dimensional, short-sequence settings ($C \gg L$) due to its purely linear channel dependence, such data structures are rare in long-term time series forecasting tasks. Meanwhile, MSTN-Transformer scaling as $\mathcal{O}(CL^2 + C^2 + CK)$ optimizes practical runtime by confining the quadratic term to a single encoder layer rather than a deep stack (N layers) as seen in iTransformer or standard Transformer architectures. This strategic design allows MSTN to balance the high-capacity modeling of Transformers with the latency requirements of Edge-AI, delivering SOTA performance with minimized computational overhead.

Table 7: Comprehensive evaluation of MSTN across diverse time-series benchmarks. Inference time (IT, in ms) of MSTN compared to models including SOFTS, iTransformer, TimesNet and PatchTST. **Red**/**Blue**: First/Second ranks.

(a) Comparison of MSTN with the MSE and IT metrics reported in SOFTS Han et al. (2024a) on the Traffic dataset ($L = 96$, $H = 720$).

	MSE	Perf. Gap	IT (ms)	Speed Gap
MSTN-Transformer	**0.020**	**1.00×**	**0.72**	**1.00×**
MSTN-BiLSTM	**0.087**	**4.30× ↓**	**3.20**	**4.44× ↓**
SOFTS	0.447	22.4× ↓	25.00	34.7× ↓
iTransformer	0.467	23.4× ↓	35.00	48.6× ↓
PatchTST	0.484	24.2× ↓	50.00	69.4× ↓
TS-Mixer	0.569	28.5× ↓	20.00	27.8× ↓
DLinear	0.645	32.3× ↓	10.00	13.9× ↓
Crossformer	0.589	29.5× ↓	150.00	208.3× ↓
Stationary	0.653	32.7× ↓	50.00	69.4× ↓
TimesNet	0.640	32.0× ↓	220.00	305.6× ↓
FEDformer	0.626	31.3× ↓	300.00	416.7× ↓

(b) Comprehensive evaluation of MSTN-Transformer across diverse time-series forecasting benchmarks Wu et al. (2022).

Dataset	Horizon	Size (MB)	IT (ms)
Traffic	96,192,336,720	2.87	0.72
ETTh2	96,192,336,720	3.65	0.91
ETTm2	96,192,336,720	2.92	0.85
Weather	96,192,336,720	2.85	0.70
Exchange	96,192,336,720	2.96	0.80
ILI	24,36,48,60	2.51	0.63

(c) IT of MSTN variants compared to TimesNet and PatchTST across seven international datasets.

Dataset	MSTN	Size (MB)	IT (ms)	TimesNet IT	PatchTST IT
Rodegast Rodegast et al. (2024a)	MSTN-Trans.	0.54	**0.15**	5.07	4.80
	MSTN-BiLSTM	4.39	**0.81**		
Boubezoul Boubezoul et al. (2020)	MSTN-Trans.	0.96	**0.24**	4.20	4.68
	MSTN-BiLSTM	4.16	**2.79**		
UCI-HAR Reyes-Ortiz et al. (2013)	MSTN-Trans.	29.98	**3.48**	45.30	4.36
	MSTN-BiLSTM	7.14	**2.03**		
PAMAP2 Reiss (2012)	MSTN-Trans.	1.58	**0.20**	0.46	3.21
	MSTN-BiLSTM	0.76	**0.086**		
ActBeCalf Dissanayake et al. (2025)	MSTN-Trans.	0.89	**0.14**	8.53	2.20
	MSTN-BiLSTM	4.14	**1.76**		
MetroPT3 Davari et al. (2021a)	MSTN-Trans.	1.30	**0.19**	0.09	0.19
	MSTN-BiLSTM	0.57	**0.042**		
NASA Saxena & Goebel (2008)	MSTN-Trans.	1.31	**0.45**	4.52	2.31
	MSTN-BiLSTM	4.25	**1.48**		

One thing to note is that despite the MSTN-BiLSTM's better theoretical scaling ($\mathcal{O}(L)$) than the MSTN-Transformer's $\mathcal{O}(L^2)$, the latter runs significantly faster than the former (Table 7a). This phenomenon is due to the fundamental conflict between the model's structure and the way modern acceleration architectures process the data. The BiLSTM encoder is inherently recurrent, meaning that the calculation of the hidden state h_t is strictly dependent on the previous state h_{t-1}. This sequential data dependency prevents the hardware's parallel processing units from being fully utilized, forcing the computation into a slow, iterative loop (low hardware utilization). In contrast, the Transformer executes its $\mathcal{O}(L^2)$ self-attention via highly optimized Vaswani et al. (2023), complete matrix parallelization. The core operation, the QK^T matrix multiplication, is performed simultaneously across thousands of processing elements. This high utilization of the hardware's parallel capacity, confined to a single encoder layer before pooling, ensures the Transformer's execution speed drastically outweighs the theoretical linearity of the BiLSTM's sequential cost, resulting in lower practical latency across high-throughput platforms.

Standard Transformers maintain the full sequence length L across a multi-layer stack of N blocks for self-attention operations, resulting in cumulative attention cost proportional to $N \cdot \mathcal{O}(CL^2)$. In contrast,

MSTN applies attention in its encoder, then immediately pools ($L \to 1$). Standard Transformers maintain L across all layers, while MSTN eliminates L-dependence after encoding. Consequently, the remainder of the network operates on static feature vectors with $\mathcal{O}(1)$ complexity relative to L for operations involving sequence length. By eliminating the recursive application of attention across deep layers, MSTN reduces the sequence-dependent FLOPs significantly. While MSTN-BiLSTM avoids quadratic attention costs entirely by employing BiLSTM encoding with $\mathcal{O}(CL)$ complexity. When combined with the remaining pipeline components (CNN, refinement, and projection), the total complexity is $\mathcal{O}(CL + C^2 + CK)$, which scales linearly with sequence length L.

Following the baseline results and inference time metrics reported in SOFTS Han et al. (2024a) on the Traffic data set (lookback window $L = 96$, horizon $H = 720$), we compare our MSTN results with their metrics reported in Table 7a. MSTN-Transformer demonstrates that architectural efficiency can surpass asymptotic predictions. MSTN-Transformer achieves 69× faster inference than PatchTST (0.72 ms vs. 50.00 ms). Furthermore, it outperforms models with better theoretical scaling, showing speedups of 34.7× over SOFTS and 305× over TimesNet. This practical gain, irrespective of the asymptotic class, is primarily delivered by the early temporal pooling of the MSTN architecture ($L \to 1$), a strategic design choice that dramatically reduces the overall FLOPs compared to deep, unpooled attention stacks. This performance represents a 22.4× accuracy improvement and 34.7× speedup over the SOFTS baseline, and a significant speed and accuracy advantage over all other models detailed in Table 7a. This improved efficiency enables substantially reduced memory usage and inference latency, positioning MSTN as a practical solution for deployment on constrained hardware platforms.

As summarized in Table 7b, MSTN demonstrates consistently efficient inference across six widely used long-term forecasting benchmarks. Despite the variation in dataset scale and temporal structure, MSTN maintains a compact model size in the range of 2.5–3.7 MB and sustains sub-millisecond latency across all horizons, including the longer 336 and 720-step settings. This stability in runtime behavior across heterogeneous datasets such as Traffic, ETTh2, ETTm2, Weather, Exchange, and ILI highlights the robustness of MSTN's dual-path, multi-scale feature extraction. The uniformly low inference cost emphasizes its suitability for real-time forecasting scenarios and deployment in resource-constrained environments where computational consistency is critical.

The inference analysis in Table 7c shows that MSTN maintains consistently low latency across seven heterogeneous datasets while maintaining a small footprint (0.54-7.14 MB). On Rodegast dataset, MSTN-Transformer (0.15 ms) achieves a 33× speedup over TimesNet (5.07 ms) and a 31× speedup over PatchTST (4.80 ms). Similar trends are observed on Boubezoul and ActBeCalf datasets. These findings highlight the scalability and deployment efficiency of MSTN across both low-channel and high-dimensional sensor datasets, underscoring its suitability for latency-sensitive and edge-oriented time-series applications.

Finally, in terms of comparative complexity, MSTN shows an advantage over the current SOTA frameworks. While TIME-LLM Jin et al. (2024) relies on a massive 6.6 billion parameter foundation model and EMTSF Alharthi et al. (2025) design utilizes 9.66 million parameters. The proposed MSTN-BiLSTM and MSTN-Transformer variants utilize fixed cores of only ∼0.40 million and ∼1.04 million parameters. This reduction in parameters compared to other models leads to better deployability of Edge-AI.

5 Conclusions

This work introduces MSTN, a DL framework founded on a hierarchical multi-scale modeling principle and defined by its strategic use of ETA. MSTN integrates core components: a multi-scale convolutional encoder that constructs a hierarchical feature pyramid; a sequential modeling component for long-term global dependency modeling, empirically validated with BiLSTM and Transformer variants; and a SGF mechanism augmented with SE and MHA to enable dynamic, context-aware feature integration. Existing methods often rely on fixed-scale structural priors (e.g., patch-based tokenization, fixed frequency transformations, or frozen backbone architectures). These limitations have historically forced a compromise between scale diversity and inference speed. In contrast, our approach efficiently integrates convolutional local pattern extraction with long-term global dependency modeling, leveraging ETA via an adaptive SGF mechanism. This integrated

design enables the model to capture multi-scale temporal features while maintaining computational efficiency, a combination we demonstrate yields SOTA performance.

Across extensive evaluations, MSTN demonstrates SOTA performance on 24 of 32 benchmark datasets spanning imputation, forecasting, classification, and cross-dataset generalization. Notably, this improved performance is achieved with exceptional parameter efficiency. MSTN achieves this competitive performance with substantially fewer parameters than comparable SOTA methods: 0.40M (MSTN-BiLSTM) and 1.04M (MSTN-Transformer) parameters versus 6.6B for TIME-LLM Jin et al. (2024) and 9.66M for EMTSF Alharthi et al. (2025), representing 16,500× and 9.3× reductions, respectively. MSTN-Transformer achieves a 22.4× reduction in prediction error (MSE) over the SOFTS model while being 34.7× faster in inference, all within a compact footprint, while MSTN-BiLSTM shows competitive performance. This dual advancement in both performance and efficiency is a distinguishing feature from prior work. This outcome, particularly the Transformer's better speed despite its quadratic complexity, validates the MSTN's ETA strategy, showing that maximizing hardware parallelization in the encoding stage is often more critical for latency than maintaining theoretical linear complexity. The model also exhibits better cross-domain generalization, with its small model size (0.54–7.14 MB) making it suitable for edge deployment. Ablation studies confirm that each component is critical, with the Transformer/BiLSTM sequence modeling core providing a foundational capability and the multi-scale and fusion modules delivering significant competitive gains.

Although MSTN excels in standard forecasting regimes ($L \gg C$), we acknowledge that factorization-based models such as SOFTS retain a theoretical efficiency advantage in high-dimensional and short-sequence settings ($C \gg L$). However, since such datasets are rarely encountered in long-term time series forecasting tasks, this limitation has a negligible impact on practical deployability.

Future work will explore cross-variate attention mechanisms and large-scale pre-training to further enhance the capabilities of MSTN as a foundation model for time series understanding. Our work paves the way for efficient multi-scale temporal modeling under resource constraints, enabling real-time applications across diverse domains from healthcare to industrial monitoring.

Data and Code Availability

All datasets used in this study are publicly available from their respective sources as cited throughout the paper. The code supporting the results presented in this paper will be made publicly available upon publication.

Competing Interests

The authors declare no conflict of interest.

References

Musleh Alharthi, Kaleel Mahmood, Sarosh Patel, and Ausif Mahmood. Emtsf:extraordinary mixture of sota models for time series forecasting, 2025. URL https://arxiv.org/abs/2510.23396.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018, 2018. URL https://arxiv.org/abs/1811.00075.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271, 2018. URL http://arxiv.org/abs/1803.01271.

Abderrahmane Boubezoul, Fabien Dufour, Samir Bouaziz, and Stéphane Espié. Corrigendum to "dataset on powered two wheelers fall and critical events detection". *Data in Brief*, 30:105577, 2020. ISSN 2352-3409.

Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters. *ACM Trans. Intell. Syst. Technol.*, 16(3), April 2025. ISSN 2157-6904. doi: 10.1145/3719207. URL https://doi.org/10.1145/3719207.

Si-An Chen, Chun-Liang Li, Sercan O Arik, Nathanael Christian Yoder, and Tomas Pfister. TSMixer: An all-MLP architecture for time series forecast-ing. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=wbpxTuXgm0.

Narjes Davari, Bruno Veloso, Rita Ribeiro, and Joao Gama. MetroPT-3 Dataset. UCI Machine Learning Repository, 2021a. DOI: https://doi.org/10.24432/C5VW3R.

Narjes Davari, Bruno Veloso, Rita P. Ribeiro, Pedro Mota Pereira, and João Gama. Predictive maintenance based on anomaly detection using deep learning for air production unit in the railway industry. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10, 2021b. doi: 10.1109/DSAA53316.2021.9564181.

Difan Deng and Marius Lindauer. Optimizing time series forecasting architectures: A hierarchical neural architecture search approach. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Ym2wqojm4e.

Oshana Iddi Dissanayake, Sarah E. McPherson, Joseph Allyndrée, Emer Kennedy, Pádraig Cunningham, and Lucile Riaboff. Actbecalf: Accelerometer-based multivariate time-series dataset for calf behavior classification. *Data in Brief*, 60:111462, 2025. ISSN 2352-3409. doi: https://doi.org/10.1016/j.dib.2025.111462. URL https://www.sciencedirect.com/science/article/pii/S2352340925001945.

Luo donghao and wang xue. ModernTCN: A modern pure convolution structure for general time series analysis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vpJMJerXHU.

Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam H. Nguyen, Wesley M. Gifford, Chandra Reddy, and Jayant Kalagnanam. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series, 2024. URL https://arxiv.org/abs/2401.03955.

Fatema Elwy, Raafat Aburukba, A. R. Al-Ali, Ahmad Al Nabulsi, Alaa Tarek, Ameen Ayub, and Mariam Elsayeh. Data-driven safe deliveries: The synergy of iot and machine learning in shared mobility. *Future Internet*, 15(10), 2023. ISSN 1999-5903.

Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4652–4663, 2019.

Aleksandra Gruca, Federico Serva, Llorenç Lliso, Pilar Rípodas, Xavier Calbet, Pedro Herruzo, Jiří Pihrt, Rudolf Raevskyi, Petr Šimánek, Matej Choma, et al. Weather4cast at neurips 2022: Super-resolution rain movie prediction under spatio-temporal shifts. In *NeurIPS 2022 Competition Track*, pp. 292–313. PMLR, 2022.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion, 2024a. URL https://arxiv.org/abs/2404.14197.

Wenyong Han, Tao Zhu, Liming Chen, Huansheng Ning, Yang Luo, and Yaping Wan. Mcformer: Multivariate time series forecasting with mixed-channels transformer. *IEEE Internet of Things Journal*, 11(17):28320–28329, 2024b. doi: 10.1109/JIOT.2024.3401697.

Yangdong He and Jiabao Zhao. Temporal convolutional networks for anomaly detection in time series. *Journal of Physics: Conference Series*, 1213(4):042050, 2019. doi: 10.1088/1742-6596/1213/4/042050.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Dewi Pramudi Ismi, Shireen Panchoo, and Murinto Murinto. K-means clustering based filter feature selection on high dimensional data. *International Journal of Advances in Intelligent Informatics*, 2:38–45, 2016. URL https://api.semanticscholar.org/CorpusID:43897444.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models, 2024. URL https://arxiv.org/abs/2310.01728.

Akhil Kadiyala and Ashok Kumar. Multivariate time series models for prediction of air quality inside a public transportation bus using available software. *Environmental Progress & Sustainable Energy*, 33(2): 337–341, 2014.

E. G. Kardakos, M. C. Alexiadis, S. I. Vagropoulos, C. K. Simoglou, P. N. Biskas, and A. G. Bakirtzis. Application of time series and artificial neural network models in short-term forecasting of pv power generation. In *2013 48th International Universities' Power Engineering Conference (UPEC)*, pp. 1–6, 2013. doi: 10.1109/UPEC.2013.6714975.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Olaf Köllé. Wetterstation. weather. Technical report and dataset, Max-Planck-Institut für Biogeochemie (BGC Jena), Germany, 2025. URL https://www.bgc-jena.mpg.de/wetter/.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks, 2018. URL https://arxiv.org/abs/1703.07015.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021. doi: 10.1098/rsta.2020.0209.

Yijing Liu, Qinxian Liu, Jian-Wei Zhang, Haozhe Feng, Zhongwei Wang, Zihan Zhou, and Wei Chen. Multivariate time-series forecasting with temporal polynomial graph neural networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=pMumil2EJh.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting, 2024. URL https://arxiv.org/abs/2310.06625.

Mohammad Amin Morid, Olivia R. Liu Sheng, and Joseph Dunbar. Time series prediction using deep learning methods in healthcare. 14(1), January 2023. ISSN 2158-656X. doi: 10.1145/3531326. URL https://doi.org/10.1145/3531326.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers, 2023. URL https://arxiv.org/abs/2211.14730.

Evandro S. Ortigossa and Eran Segal. Seg-moe: Multi-resolution segment-wise mixture-of-experts for time series forecasting transformers, 2026. URL https://arxiv.org/abs/2601.21641.

Xiangfei Qiu, Hanyin Cheng, Xingjian Wu, Jilin Hu, Chenjuan Guo, and Bin Yang. A comprehensive survey of deep learning for multivariate time series forecasting: A channel strategy perspective, 2025. URL https://arxiv.org/abs/2502.10721.

Attila Reiss. PAMAP2 Physical Activity Monitoring. UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C5NW2H.

Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th International Symposium on Wearable Computers*, pp. 108–109, 2012. doi: 10.1109/ISWC.2012.13.

Jorge Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto, and Xavier Parra. Human Activity Recognition Using Smartphones. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C54S4K.

M. Rodegast et al. Motorcycle collision dataset, 2024a. URL https://darus.uni-stuttgart.de/dataset.xhtml?persistentId=doi:10.18419/darus-3301.

Philipp Rodegast, Steffen Maier, Jonas Kneifl, and Jörg Fehr. On using machine learning algorithms for motorcycle collision detection. *Discover Applied Sciences*, 6(6):326, 2024b. ISSN 3004-9261.

Abhinav Saxena and Kai Goebel. Nasa turbofan engine degradation simulation data set, 2008. URL https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/. NASA Ames Prognostics Center of Excellence.

Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series, 2021. URL https://arxiv.org/abs/2101.06861.

Artur Trindade. ElectricityLoadDiagrams20112014. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C58C86.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting, 2022. URL https://arxiv.org/abs/2202.01381.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, 2022. URL https://arxiv.org/abs/2106.13008.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis, 2023. URL https://arxiv.org/abs/2210.02186.

Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective, 2023. URL https://arxiv.org/abs/2311.06190.

Wenzhen Yue, Ruohao Guo, Ji Shi, Zihan Hao, Shiyu Hu, and Xianghua Ying. vlinear: A powerful linear model for multivariate time series forecasting, 2026. URL https://arxiv.org/abs/2601.13768.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting?, 2022. URL https://arxiv.org/abs/2205.13504.

Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures, 2022. URL https://arxiv.org/abs/2207.01186.

Yitian Zhang, Liheng Ma, Soumyasundar Pal, Yingxue Zhang, and Mark Coates. Multi-resolution time-series transformer for long-term forecasting, 2024. URL https://arxiv.org/abs/2311.04147.

Shubao Zhao, Ming Jin, Zhaoxiang Hou, Chengyi Yang, Zengxiang Li, Qingsong Wen, and Yi Wang. Himtm: Hierarchical multi-scale masked time series modeling with self-distillation for long-term forecasting, 2024. URL https://arxiv.org/abs/2401.05012.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting, 2022a. URL https://arxiv.org/abs/2201.12740.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting, 2022b. URL https://arxiv.org/abs/2201.12740.

Tian Zhou, PeiSong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all:power general time series analysis by pretrained lm, 2023. URL https://arxiv.org/abs/2302.11939.