MSTN: Fast and Efficient Multivariate Time Series Model

Sumit S Shevtekar[a,*], Chandresh K Maurya[a], Gourab Sil[b]

[a] *Department of Computer Science and Engineering, Indian Institute of Technology Indore, Indore, 452020, Madhya Pradesh, India*
[b] *Department of Civil Engineering, Indian Institute of Technology Indore, Indore, 452020, Madhya Pradesh, India*

Abstract

Real-world time-series data is highly non stationary and complex in dynamics that operate across multiple timescales, ranging from fast, short-term changes to slow, long-term trends. Most existing models rely on fixed-scale structural priors, such as patch-based tokenization, fixed frequency transformations, or frozen backbone architectures. This often leads to over-regularization of temporal dynamics, which limits their ability to adaptively model the full spectrum of temporal variations and impairs their performance on unpredictable, Sudden, high-magnitude events. To address this, we introduce the Multi-scale Temporal Network (MSTN), a novel deep learning architecture founded on a hierarchical multi-scale and sequence modeling principle. The MSTN framework integrates three core components: (i) a multi-scale convolutional encoder that constructs a hierarchical feature pyramid for local patterns (ii) a sequence modeling component for long-range temporal dependencies. We empirically validate this with Bidirectional Long Short-Term Memory (LSTM) and Transformer variants, establishing a flexible foundation for future architectural advancements. and (iii) a gated fusion mechanism augmented with squeeze-and-excitation (SE) and multi-head temporal attention (MHTA) for dynamic, context-aware feature integration. This design enables MSTN to adaptively model temporal patterns from milliseconds to long-range dependencies within a unified framework. Extensive evaluations across time-series long-horizon forecasting, imputation, classification and generalizability study demonstrate that MSTN achieves competitive state-of-the-art (SOTA) performance, showing improvements over contemporary approaches including EMTSF, LLM4TS, HiMTM, TIME-LLM, MTST, SOFTS, iTransformer, TimesNet, and PatchTST. In total, MSTN establishes new SOTA performance on 24 of 32 benchmark datasets, demonstrating its consistent performance across diverse temporal tasks.

Keywords:
Multivariate time series, multi-scale temporal modeling, time series forecasting,

[*] Corresponding author Email: sumit.shevtekar@gmail.com

classification, imputation, computational efficiency, deep learning architectures

1. Introduction

Multivariate time series analysis is a crucial component of modern AI, with significant real-world applications ranging from healthcare monitoring and industrial predictive maintenance to behavioral analytics. It has been extensively studied for many years and has been widely used in tasks such as weather forecasting, imputing missing data for data mining, and classification of trajectories for action recognition [1, 2]. Due to its significant practical value, time series analysis has attracted considerable attention. Unlike language or video, time-series data consists of individual scalar measurements, where single time points rarely carry sufficient semantic meaning. This fundamental characteristic forces models to extract information from temporal variation patterns of continuity, periodicity, and trend, making robust temporal modeling exceptionally challenging [3, 4].

Deep learning (DL) models have demonstrated strong performance not only in forecasting tasks, but also in representation learning, where abstract features can be extracted and transferred to downstream tasks such as classification and forecasting, achieving SOTA results [5]. Recent findings show that simple linear models can outperform sophisticated Transformer variants on common benchmarks [6], highlighting fundamental limitations in contemporary architectures. This is particularly critical in safety-sensitive domains like behavioral analytics and industrial monitoring, where patterns are often non-periodic and evolve across multiple timescales. The design of temporal models faces the constant challenge of balancing model strength with computational efficiency. Models like recurrent neural networks (RNN) and their gated variants [7, 8] offer a strong sequential modeling but suffer from vanishing gradients and limited parallelism. Temporal convolutional networks (TCN) [9] improve computational efficiency but are constrained by fixed receptive fields. The Transformer architecture [10] originally promised global dependency modeling, but its quadratic complexity prompted a line of research focused primarily on sparsification. Models like Informer [11] and FEDformer [12] develop efficient attention variants, yet often retain a point-wise view of time series, treating each time step as an independent token and struggling to capture local semantic structures.

These challenges have shaped several predominant, yet ultimately constrained, modern paradigms. One approach, exemplified by PatchTST [5], adapts the patching technique from Vision Transformers, grouping adjacent time points into sub-series tokens. This approach successfully captures local semantics and benefits from channel-independent processing, but its reliance on a fixed patch size creates a fundamental rigidity, locking the model into a single, pre-defined scale of analysis. Another paradigm, improved by TimesNet [3], reframes the problem by transforming 1D time series into 2D tensors based on identified periods. While powerful for data with strong periodic components, this approach is intrinsically dependent on reliable periodicity detection and the 2D transformation prior, making it less suited for non-stationary, irregular, or non-periodic patterns where such clear structure is absent. Building on these concepts, MTST [13] employs a multi-resolution patch-based design with a multi-branch architecture for simultaneous modeling of diverse temporal patterns at different resolutions. However, its fixed choice of resolutions is less flexible and may fail to generalize across datasets with irregular or heterogeneous temporal patterns. Patch tokenization can also lose fine-grained dynamics, while the reliance on relative positional encoding restricts robustness in non-seasonal or event-driven sequences.

We identify a fundamental limitation on architectures that impose specific single-scale structural constraints, such as a fixed patch size, a period-based 2D shape, or frozen backbone architectures. These prior models struggle with the inherently multi-scale nature of many real-world temporal incidents. Their architecture limits their ability to model discriminative patterns that manifest simultaneously across different timescales, from brief micro-events to slow macro and long-term trends, such as found in risky driving behaviour. This gap highlights the need for a more flexible temporal representation that can primarily adapt to this hierarchical structure without relying on a single, potentially misaligned, structural assumption.

To address these challenges, we propose a DL model (MSTN) that naturally captures temporal hierarchies without relying on fixed and repeated structural assumptions. MSTN unifies multi-scale feature learning with global sequence modeling through a dual-path architecture with a sequence modeling core. We empirically validate this design with two instantiations: i) MSTN–BiLSTM, which excels at modeling local continuity and behavioral smoothness, and ii) MSTN–Transformer, which captures local complex patterns and long-range contextual dependencies. In both variants, convolutional encoders extract fine-grained local features, while the temporal core captures long-range dependencies. Their outputs are integrated via a gated fusion mechanism enhanced with SE recalibration and MHTA, yielding compact, discriminative, and temporally coherent representations. This design provides an unprecedented trade-off between representational capacity and computational efficiency.

We demonstrate that MSTN establishes new SOTA performance on 24 of 32 benchmark datasets for time series analysis. This includes 9 forecasting benchmarks, and 6 imputation tasks and 17 classification datasets (10 from the standard UEA archive and 7 additional international benchmarks), demonstrating its versatility and improved performance across diverse applications. MSTN establishes a new SOTA in long-term forecasting, attaining the lowest prediction error on 6 out of 9 standard benchmarks and consistently exceeds leading architectures, including EMTSF [14], LLM4TS [15], HiMTM [2], TIME-LLM [16], MTST [13], GPT2(6) [17], SOFTS [18] iTransformer [19], and PatchTST [5]. Complementing these results, the MSTN-Transformer variant achieves first-place rankings 31 out of 48 (≈65%) in all 48 imputation scenarios across the ETT, Electricity, and Weather datasets, demonstrating strong resilience under up to 50% missingness. evaluation scenarios and secure the highest average score on 5 out of 6 datasets. In the classification task, MSTN achieves improved performance over recent SOTA architectures such as GTP2(6) [17], TimesNet [3], LigthtTS [20] and FEDformer [12]. We further evaluate MSTN for generalizability across 7 benchmark datasets from various domains like human safety, physical activity recognition, animal-welfare monitoring, and mechanical prognostics. Without domain-specific tuning, surprisingly, MSTN achieves uniformly strong performance, ranks first across all datasets, establishing new benchmarks and illustrating its robustness under diverse temporal dynamics. Finally, MSTN-Transformer achieves unprecedented performance with 0.020 MSE, 2.87 MB footprint and 0.72 ms inference time, representing a 22.4× accuracy improvement and 34.7× speedup over SOFTS [18]. The model further demonstrates improved classification accuracy (99.53%) with ultra-low latency (0.155 ms) significantly outperforming TimesNet (99.18%, 5.07 ms) and PatchTST (98.64%, 4.80 ms) on the Rodegast et al. [21] dataset. While maintaining compact model sizes (0.54–7.14 MB), this balance of SOTA accuracy, robustness, and efficiency positions MSTN-Transformer as an ideal solution for real-time and edge-AI temporal modeling applications. Our comprehensive evaluation establishes MSTN as a new benchmark, achieving SOTA performance on 24 of 32 datasets across forecasting, imputation, and classification tasks.

2. Results

This section presents a comprehensive evaluation of the proposed MSTN. Demonstrating its SOTA performance in the long-term forecasting, imputation, classification, and generalizability study. MSTN (MSTN-Transformer and MSTN-BiLSTM) achieves competitive performance, showing improvements over recent DL, Transformer based, and time series approaches.

2.1. Long-Term Forecasting

We evaluate long-term forecasting performance using established protocols from the prior literature [1, 3, 5] 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 1, the MSTN-Transformer and MSTN BiLSTM shows improved performance compared to specialized time series forecasting architectures, including EMTSF [14], LLM4TS [15], HiMTM [2], TIME-LLM [16], MTST [13],SOFTS [18], and iTransformer [19].

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 places while showing improved performance over all other baselines. On the Traffic dataset, MSTN-Transformer achieves an average MSE of 0.019 (19.9× lower than EMTSF) while MSTN-BiLSTM delivers competitive 0.086 MSE. Similarly,

Table 1: Multivariate forecasting results with prediction lengths $H \in \{96, 192, 336, 720\}$ for others and for ILI $H \in \{24, 36, 48, 60\}$. **Red**/**Blue**: First/Second ranks.

Models	MSTN-Trans.		MSTN BiL.		EMTSF		LLM4TS		HiMTM		TIME-LLM		MTST		SOFTS		iTransf.	
	Ours		Ours		2025		2025		2024		2024		2024		2024		2024	
H-Metric	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**	0.285	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	0.324	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	0.353	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	0.408	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	0.342	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**	0.165	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**	0.220	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**	**0.268**	**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**	**0.350**	**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**	0.251	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**	0.371	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**	0.403	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	0.420	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	0.422	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	0.404	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**	0.269	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**	0.328	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**	0.353	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**	0.383	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**	0.333	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**	0.128	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**	0.146	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**	0.163	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**	0.200	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**	0.159	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	0.372	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	0.391	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	0.405	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	0.437	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	0.401	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**	0.147	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**	0.191	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**	0.241	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	0.313	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**	0.223	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	-	-	-	-	-	-	-	-	-	-	-	-	-	-
192	0.456	0.566	0.278	0.437	-	-	-	-	-	-	-	-	-	-	-	-	-	-
336	0.906	0.803	0.303	0.469	-	-	-	-	-	-	-	-	-	-	-	-	-	-
720	0.815	0.734	0.818	0.795	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Avg	0.668	0.675	0.408	0.526	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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	-	-	-	-	-	-

on ECL dataset, MSTN-Transformer's 0.043 MSE (3.6× better than EMTSF) and MSTN-BiLSTM shows robust performance. Both MSTN variants dominate the ILI dataset with MSEs of 0.165 and 0.166, outperforming TIME-LLM by 8.7×. MSTN-BiLSTM particularly excels on Exchange dataset with 0.408 average MSE, demonstrating 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.

2.2. Imputation

Table 2: 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.

Models	MSTN-Tra. (Ours)		MSTN BiL. (Ours)		GPT2(3) (2023)		TimesNet (2023)		PatchTST (2023)		LightTS (2022)		DLinear (2023)		Stationary (2022)	
Mask R.	**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	0.060	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	0.080	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	0.102	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	0.133	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	0.094	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	0.042	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	0.049	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	0.056	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	0.065	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	0.053	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	0.026	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	0.032	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	**0.039**	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	0.047	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	0.036	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	0.021	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	**0.024**	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	0.027	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	0.030	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	0.026	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	0.093	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	0.097	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	0.102	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	0.108	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	0.100	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	0.027	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	0.029	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	0.033	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	0.037	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	0.032	0.059

As presented in Table 2, 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 [3], we evaluate imputation performance on electricity and weather domain datasets, including ETT [11], ECL [22], and Weather [23]. To simulate real-world scenarios, we employ random masking ratios of 12.5%, 25%, 37.5%, 50%. This shows improved performance over several recent approaches, including GPT2(3) [17], TimesNet [3], PatchTST [5], LightTS [20], and DLinear [6].

2.3. Time-Series Classification

Table 3a presents the classification results across 10 UEA standard benchmark datasets. The proposed MSTN framework is compared against SOTA time-series classification methods, including GPT-2(6) [17], TimesNet [3], FEDformer [12], and LightTS [20]. MSTN demonstrates competitive performance in multiple datasets. MSTN-Transformer achieves perfect classification (100%) on JapaneseVowels and near-perfect performance on SpokenArabic (99.32%), while MSTN-BiLSTM achieves improved accuracy on Heartbeat (81.49%), JapaneseVowels (100%) and PEMS-SF (96.59%).

Table 3: 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**	32.43	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	63.20	60.30	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	**43.0**	35.50	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	78.95	**81.49**	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	**100**	**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	**94.32**	**96.59**	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	**93.79**	90.27	**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	**62.42**	**60.53**	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	99.32	99.15	99.2	99.0	**100**	81.4	98.8	**100**	**100**	**100**	**100**	**99.6**	**100**	97.0	98.4
UWave	76.00	68.18	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

(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.**
Rodeg.[21]	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][24]
				Transf.	**99.53**	0.947	0.951	0.942	0.54	**0.155**					
Boubez.[25]	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][26]
				Transf.	91.76	0.927	0.921	0.934	0.96	**0.24**					
UCI-HAR[27]	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][28]
				Transf.	**95.58**	0.956	0.958	0.955	29.98	**3.48**					
PAMAP2[29]	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][30]
				Transf.	**99.52**	0.995	0.995	0.995	1.58	**0.20**					
ActBeC.[31]	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][31]
				Transf.	88.14	0.883	0.873	0.913	0.89	**0.14**					
MetroPT3[32]	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][33]
				Transf.	**93.83**	0.938	0.940	0.938	1.30	**0.19**					
NASA[34]	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.

2.4. Generalizability Study

As summarized in Table 3b, the proposed MSTN framework demonstrates remarkable 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 [3] and PatchTST [5] in both accuracy and computational efficiency.

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[21], substantially outperforming prior work (91.0% RF,GB [24]). For activity recognition, MSTN-BiLSTM achieves 96.59% accuracy on the UCI-HAR dataset[27], 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[31] assessment with 93.00% accuracy on calf behavior recognition. Furthermore, MSTN-Transformer demonstrates exceptional capability in mechanical prognostics, achieving 11.26 RMSE on the NASA Turbofan dataset[34] and significantly outperforming all baseline predictors in remaining useful life estimation.

MSTN delivers this performance with good computational efficiency. For instance, on the Rodegast [21] Germany data set, the MSTN-Transformer variant achieves 0.155 ms/sample inference latency 33× faster than TimesNet and 31.0× faster than PatchTST while maintaining compact model size. This efficiency enables deployment on resource-constrained edge devices for real-time safety applications. MSTN achieves consistent SOTA performance across fundamentally different domains from behavioral analytics to mechanical systems, without any architectural modifications.

2.5. Ablation Study

Table 4 presents an ablation study evaluating the effectiveness of the proposed MSTN-Transformer components across six datasets (ECL, Weather, ETTm1, ETTh2, Traffic, and ILI) and the MSTN-BiLSTM architecture components on the Weather, ETTm2, and Traffic datasets. To verify the contribution of each component, we compare the full MSTN-Transformer model against five variants: w/o CNN, which removes the convolutional pathway; w/o Transformer, which eliminates the core Transformer module; w/o SE, which removes the squeeze-and-excitation blocks; w/o MHTA, which excludes the Multi-Head Temporal Attention; and w/o Gated F., which removes the gated fusion mechanism. In MSTN-BiLSTM architecture, w/o BiLSTM denotes the variant of MSTN-BiLSTM without the BiLSTM layer to assess its importance in temporal sequence modeling. These results validate our architectural design choices and demonstrate the complementary nature of the integrated components.

2.6. Model Complexity and Edge-AI Deployability

The MSTN-Transformer and MSTN-BiLSTM architectures exemplify a fundamental design trade-off in temporal modeling. Both architectures integrate efficient parallel components CNN pathways ($\mathcal{O}(T)$) with either Transformer ($\mathcal{O}(T^2)$) or BiLSTM ($\mathcal{O}(T)$) mechanisms, combined with $\mathcal{O}(T)$ fusion operations and finalized by a multi-head temporal attention layer of $\mathcal{O}(T^2)$ complexity. This quadratic component dictates the overall asymptotic complexity of $\mathcal{O}(T^2)$. However, this design preserves robust temporal modeling capabilities and achieves improved performance and practical efficiency compared to pure $\mathcal{O}(T^2)$ Transformer-based models. Linear models such as DLinear and TS-Mixer achieve optimal $\mathcal{O}(T)$ complexity, while spectral and frequency-based

Table 4: Ablation study of MSTN-Transformer and MSTN-BiLSTM. $H \in \{96, 192, 336, 720\}$ and for ILI $H \in \{24, 36, 48, 60\}$.

Methods	**MSTN Full**	**w/o CNN**	**w/o Transf.**	**w/o SE**	**w/o MHTA**	**w/o Gated F.**
Metric	**MSE MAE**	**MSE MAE**	**MSE MAE**	**MSE MAE**	**MSE MAE**	**MSE MAE**
MSTN Transformer						
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	0.043 0.165	0.043 0.169	0.048 0.173	0.043 0.164	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 0.172	0.050 0.178	0.053 0.179
Weather						
96	0.109 0.263	0.111 0.265	0.118 0.273	0.109 0.263	0.110 0.264	0.110 0.264
192	0.111 0.264	0.115 0.269	0.120 0.275	0.111 0.264	0.111 0.265	0.112 0.265
336	0.114 0.268	0.120 0.274	0.123 0.279	0.113 0.268	0.114 0.269	0.114 0.269
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	0.055 0.175	0.089 0.227	0.061 0.185	0.058 0.178	0.055 0.175
192	0.103 0.242	0.105 0.243	0.105 0.243	0.109 0.246	0.098 0.233	0.096 0.229
336	0.127 0.271	0.129 0.272	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	0.365 0.485	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	0.445 0.529	0.382 0.498	0.395 0.501
336	0.377 0.495	0.415 0.520	0.388 0.501	0.396 0.504	0.387 0.498	0.396 0.505
720	0.400 0.509	0.427 0.522	0.384 0.498	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	0.021 0.116
336	0.020 0.112	0.022 0.118	0.029 0.135	0.021 0.115	0.024 0.118	0.023 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	0.132 0.256	0.131 0.253	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	0.187 0.327	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	0.193 0.332	0.194 0.338
MSTN BiLSTM						
Methods	**MSTN Full**	**w/o CNN**	**w/o BiLSTM**	**w/o SE**	**w/o MHTA**	**w/o Gated F.**
Weather						
96	0.110 0.264	0.112 0.266	0.119 0.274	0.110 0.264	0.110 0.263	0.110 0.264
192	0.111 0.265	0.1132 0.267	0.120 0.275	0.112 0.265	0.111 0.265	0.112 0.266
336	0.115 0.269	0.116 0.270	0.123 0.279	0.115 0.269	0.115 0.269	0.115 0.269
720	0.132 0.289	0.138 0.292	0.139 0.296	0.132 0.293	0.133 0.295	0.134 0.298
ETTm2						
96	0.238 0.390	0.243 0.400	0.264 0.430	0.262 0.394	0.240 0.388	0.261 0.399
192	0.426 0.514	0.501 0.560	0.528 0.617	0.436 0.540	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	0.592 0.635	0.578 0.619	0.681 0.661	0.829 0.738	0.641 0.647
Traffic						
96	0.085 0.202	0.086 0.202	0.102 0.222	0.085 0.202	0.087 0.204	0.087 0.205
192	0.087 0.205	0.088 0.207	0.101 0.221	0.087 0.205	0.088 0.208	0.088 0.209
336	0.086 0.204	0.086 0.205	0.105 0.227	0.089 0.213	0.090 0.214	0.090 0.215
720	0.087 0.207	0.089 0.210	0.102 0.224	0.092 0.218	0.091 0.217	0.093 0.218

approaches such as FEDformer and TimesNet attain $\mathcal{O}(T)$ and $\mathcal{O}(T \log T)$ complexity, respectively. The recently proposed MTST further reduces the cost to $\mathcal{O}(J_{b'_n}^2)$ via patching. Although efficient, these methods often trade off capacity for unconstrained long-range dependency modeling.

Following the results and inference time metrics reported in SOFTS [18] on the Traffic data set (lookback window $L = 96$, horizon $H = 720$), we compare our results with their metrics reported in Table 5. The MSTN-Transformer achieves improved performance with 0.020 MSE and 0.72 ms inference time, representing 22.4× accuracy improvement and 34.7× speedup over SOFTS, achieving a favorable balance of accuracy and speed. This improved efficiency enables substantially reduced memory usage and inference latency, positioning MSTN as a practical solution for deployment on constrained hardware platforms.

Table 5: A: Performance Analysis (Traffic Dataset, $L = 96$, $H = 720$). B: MSTN-Trans. Inference Time Component Contributions. **Red**/**Blue**: First/Second ranks.

	MSE	**Perf. Gap**	**Inf. Time (ms)**	**Speed Gap**
A: SOTA MSE and Efficiency				
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: MSTN-Trans. Component Contributions Ablation Study (Traffic Dataset, $H = 720$)				
Variant	**Size (MB)**	**Time (ms)**	**Speed Gap**	**Role**
Full MSTN-Transformer	2.87	0.72	**1.00×**	Complete System
No Transformer	0.89	0.20	3.60× ↑	Core Layer
No CNN	2.26	0.52	1.38× ↑	Local Expert
No MHTA	2.30	0.53	1.36× ↑	Temporal Refiner
No Gated Fusion	2.73	0.62	1.16× ↑	Feature Integrator
No SE Blocks	2.85	0.65	1.11× ↑	Channel Optimizer

As shown in Table 3b, MSTN-Transformer delivers competitive performance across domains, attaining 99.53% accuracy on Rodegast [21] while maintaining efficient inference at 0.155 ms per sample with only 0.54 MB model size. This represents up to 33× speedup over TimesNet (5.07 ms) and 31.0× over PatchTST (4.80 ms), demonstrating favorable performance-efficiency trade-offs across diverse datasets. The combination of compact model sizes (0.54–7.14 MB), improved accuracy, and millisecond-level latency positions MSTN as an ideal solution for complex multivariate time series analysis tasks in edge computing environments. These results demonstrate that capable temporal models can achieve practical deployability in edge environments. The small model footprint satisfies memory constraints, millisecond-level latency enables real-time operation, and high accuracy ensures reliability. This collectively addresses core edge-AI challenges for applications like safety monitoring, predictive maintenance, and activity recognition.

3. Discussion

This work introduces MSTN, a DL architecture founded on a hierarchical multi-scale modeling principle. It integrates three core components: a multi-scale convolutional encoder that constructs a hierarchical feature pyramid; sequential modeling component for global dependency modeling, empirically validated with BiLSTM and Transformer variants; and a gated fusion mechanism augmented with SE and MHTA 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), forcing a compromise between scale diversity and inference speed. In contrast, our approach efficiently integrates convolutional local pattern extraction with global dependency modeling via an adaptive gated fusion mechanism. This enables our model to capture multi-scale temporal features without compromising computational efficiency, ultimately achieving state-of-the-art performance.

Comprehensive evaluations demonstrate that the MSTN-Transformer establishes a new SOTA across forecasting, imputation, classification, and generalizability. It achieves a 22.4× improvement in accuracy over the SOFTS model while being 34.7× faster in inference, all within a compact 2.87 MB footprint, while MSTN BiLSTM shows competitive performance. This dual advancement in both performance and efficiency is a key differentiator from prior work. The model also exhibits exceptional 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 core providing a foundational capability and the multi-scale and fusion modules delivering significant competitive gains.

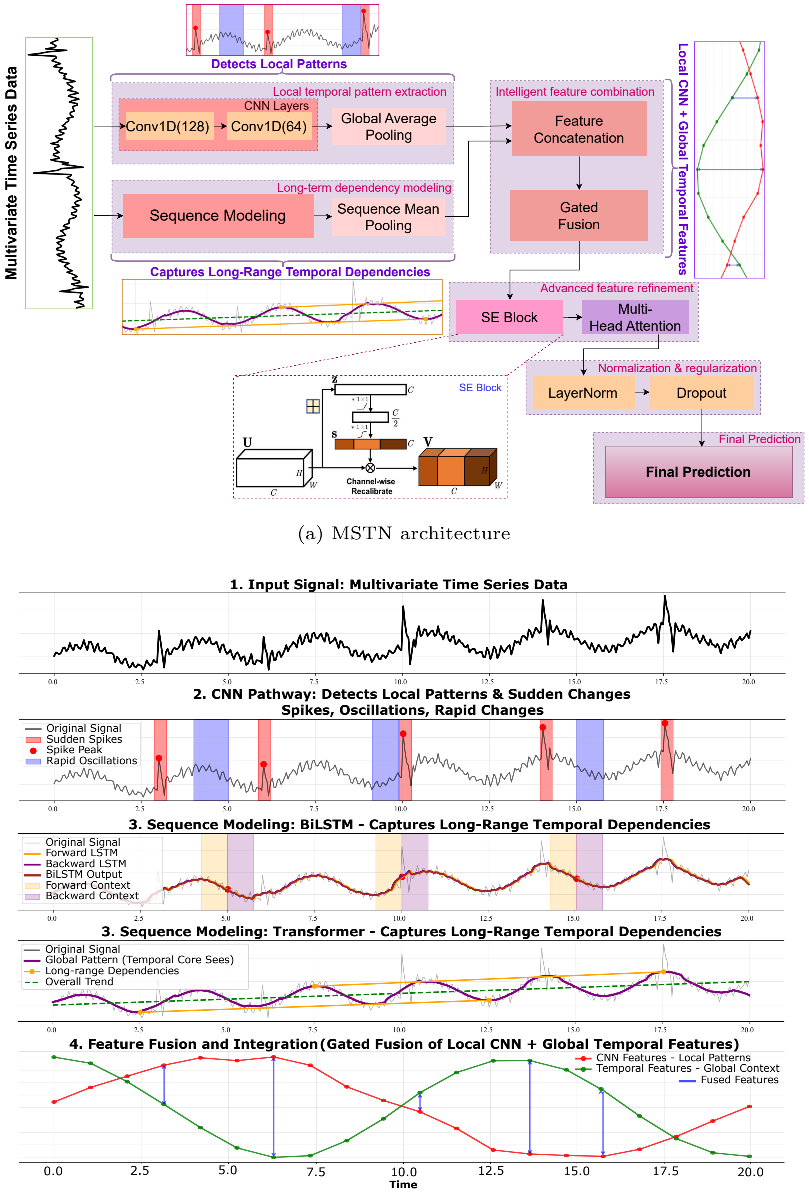
Future work will explore cross-variate attention mechanisms, and large-scale pre-training to further enhance MSTN's capabilities as a foundation model for time series understanding. Our work paves the way for efficient multi-scale temporal modeling under resource constraints, enabling real-time forecasting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

4. Methods

4.1. Proposed Architecture: MSTN

4.1.1. Architecture Overview

The proposed MSTN (Fig. 1a) is a novel hybrid DL architecture founded on a hierarchical multi-scale modeling principle. It uses a dual-branch processing pipeline: (i) a convolutional branch that extracts detailed local motion patterns, and (ii) sequence modeling core that learns temporal dependencies in the time-series data, which we empirically validate using both BiLSTM and Transformer variant in forecasting, imputation, classification and generalizability studies. (iii) The outputs of both branches are integrated using a learnable gated fusion mechanism, followed by channel-wise recalibration through SE blocks and MHTA, enabling the model to focus on critical, risk-indicating time steps. The fused representation is normalized, regularized with dropout, and linearly projected to produce probabilities. Training takes advantage of focal loss to mitigate class imbalance. Fig. 1b shows the processing pipeline of


Multivariate Time Series Data
Detects Local Patterns
Local temporal pattern extraction
CNN Layers
Conv1D(128)
Conv1D(64)
Global Average Pooling
Long-term dependency modeling
Sequence Modeling
Sequence Mean Pooling
Captures Long-Range Temporal Dependencies
Intelligent feature combination
Feature Concatenation
Gated Fusion
Local CNN + Global Temporal Features
Advanced feature refinement
SE Block
Multi-Head Attention
Normalization & regularization
LayerNorm
Dropout
Final Prediction
Final Prediction
SE Block
Z
C
∗ 1×1
C/2
∗ 1×1
S
C
U
H
W
C
Channel-wise Recalibrate
V
H
W
C
1. Input Signal: Multivariate Time Series Data
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
2. CNN Pathway: Detects Local Patterns & Sudden Changes
Spikes, Oscillations, Rapid Changes
Original Signal
Sudden Spikes
Spike Peak
Rapid Oscillations
3. Sequence Modeling: BiLSTM - Captures Long-Range Temporal Dependencies
Original Signal
Forward LSTM
Backward LSTM
BiLSTM Output
Forward Context
Backward Context
3. Sequence Modeling: Transformer - Captures Long-Range Temporal Dependencies
Original Signal
Global Pattern (Temporal Core Sees)
Long-range Dependencies
Overall Trend
4. Feature Fusion and Integration(Gated Fusion of Local CNN + Global Temporal Features)
CNN Features - Local Patterns
Temporal Features - Global Context
Fused Features
Time

(a) MSTN architecture

(b) MSTN multi-scale signal processing pipeline

Fig. 1: Proposed MSTN: (a) architectural diagram and (b) signal processing pipeline

time series signals. CNN pathway captures local patterns, while BilSTM and Transformer captures long-range temporal dependencies in the time-series data, with gated fusion adaptively combining both feature streams.

4.1.2. Model Structure

As aforementioned, based on the multi-scale nature of time series, we propose MSTN with a parallel architecture to capture temporal patterns at different resolutions. For comprehensive temporal modeling, we design dual pathways within MSTN that process the time series through complementary mechanisms and simultaneously model both local and global variations through a parameter-efficient novel hybrid block. The MSTN architecture and signal processing pipeline are illustrated in Fig. 1.

We consider the following problem: Given multivariate time series samples $X_{1:L} \in \mathbb{R}^{L \times M}$ with lookback window sequence length L and M variates features, MSTN learns rich temporal representations for multiple tasks including forecasting, imputation and classification through a novel parallel architecture.

Forward Process. The input $X \in \mathbb{R}^{B \times T \times D}$ is processed through dual complementary pathways that operate simultaneously. The CNN pathway captures local temporal patterns while the Transformer pathway models global dependencies or BiLSTM models sequential intelligence, with their outputs fused through sophisticated gated mechanisms for enhanced representation learning. To capture multi-scale temporal variations in length-T time series with D variates organized as $X_{1D} \in \mathbb{R}^{B \times T \times D}$, the architecture processes inputs through two parallel pathways with convolutional pathway and sequence modeling (Transformer or BiLSTM pathway).

The convolutional pathway captures fine-grained, short-term temporal patterns through hierarchical feature learning. The input undergoes successive 1D convolutions to extract multi-scale features, expanding the feature dimension while preserving temporal resolution, followed by refinement of the representations. To aggregate these temporal features into a compact representation, we apply global average pooling across the entire sequence:

$$H_{\text{conv}}^{(1)} = \text{ReLU}(\text{Conv1D}_7(X_{1D})) \in \mathbb{R}^{B \times 128 \times T} \tag{1}$$

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

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

where each of the 64 feature channels is summarized by its average activation over time through temporal pooling. The Conv1D_k convolutional layers employ a hierarchical design where 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.

We empirically validate two complementary sequence modeling approaches: MSTN-Transformer and MSTN-BiLSTM. Features from both pathways are elegantly concatenated, uniting the CNN's local precision with either Transformer or BiLSTM, to capture long-range temporal dependencies. When using Transformer: The architecture captures long-range dependencies through a multi-layer transformer encoder. The input is processed through self-attention mechanisms where each of the T time steps maintains a 64-dimensional representation enriched with global contextual information. To consolidate these temporal representations, we apply sequence mean pooling across the time dimension:

$$H_{\text{trans}} = \text{TransformerEnc}(X) \in \mathbb{R}^{B\times T\times 64}; \quad \mathbf{z}_{\text{trans}} = \frac{1}{T}\sum_{t=1}^{T} H_{\text{trans},t} \in \mathbb{R}^{B\times 64} \tag{4}$$

The transformer encoder uses 4 layers with 8 attention heads. When using BiLSTM: The architecture captures bidirectional sequential dependencies, processing sequences in both forward and backward directions to learn complex temporal dynamics and memorize long-range patterns:

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

The BiLSTM employs 2 layers with 64 hidden units per direction, capturing both causal and anti-causal temporal relationships. The CNN pathway utilizes global average pooling to capture local pattern summaries, while the Transformer and BiLSTM pathways employ sequence mean pooling to preserve global contextual information. The parallel features undergo sophisticated fusion through a multi-stage enhancement process that intelligently combines the strengths of both pathways. For MSTN-Transformer:

$$\mathbf{z}_{\text{concat}} = [\mathbf{z}_{\text{cnn}}; \mathbf{z}_{\text{trans}}] \in \mathbb{R}^{128} \tag{6}$$

where $\mathbf{z}_{\text{cnn}} \in \mathbb{R}^{64}$ and $\mathbf{z}_{\text{trans}} \in \mathbb{R}^{64}$. For MSTN-BiLSTM:

$$\mathbf{z}_{\text{concat}} = [\mathbf{z}_{\text{cnn}}; \mathbf{z}_{\text{bilstm}}] \in \mathbb{R}^{192} \tag{7}$$

where $\mathbf{z}_{\text{cnn}} \in \mathbb{R}^{64}$ and $\mathbf{z}_{\text{bilstm}} \in \mathbb{R}^{128}$. The architecture then performs adaptive weighting through an intelligent gated fusion mechanism, illustrated in Fig. 1, that learns to dynamically balance feature contributions:

$$\mathbf{z}_{\text{fused}} = \mathbf{z}_{\text{concat}} \odot \sigma(W_g \mathbf{z}_{\text{concat}} + b_g) \tag{8}$$

where for MSTN-Transformer: $W_g \in \mathbb{R}^{128\times 128}$, $b_g \in \mathbb{R}^{128}$ and for MSTN-BiLSTM: $W_g \in \mathbb{R}^{192\times 192}$, $b_g \in \mathbb{R}^{192}$. This mechanism learns to emphasize the most relevant features from each pathway, creating a harmonious blend of multi-scale temporal intelligence. The fused features are reshaped for sequence processing: $z_{\text{fused\_seq}} = z_{\text{fused}}.\text{unsqueeze}(1).\text{repeat}(1, T, 1) \in \mathbb{R}^{B\times T\times d}$ where $d = 128$ for MSTN-Transformer and $d = 192$ for MSTN-BiLSTM. Channel-wise attention then enhances features through separate SE blocks for each variant: For MSTN-Transformer: $z_{\text{se}} = z_{\text{fused\_seq}} \odot \sigma(W_2\text{ReLU}(W_1 \frac{1}{T}\sum_{t=1}^{T} z_{\text{fused\_seq},t}))$

where $W_1 \in \mathbb{R}^{16\times128}$, $W_2 \in \mathbb{R}^{128\times16}$ with reduction ratio 8. For MSTN-BiLSTM: $z_{\text{se}} = z_{\text{fused\_seq}} \odot \sigma(W_2\text{ReLU}(W_1\frac{1}{T}\sum_{t=1}^{T} z_{\text{fused\_seq},t}))$ where $W_1 \in \mathbb{R}^{24\times192}$, $W_2 \in \mathbb{R}^{192\times24}$ with reduction ratio 8. The SE mechanism amplifies informative channels while suppressing less useful ones. Temporal relationships are refined through multi-head attention that captures interdependencies across the enhanced features:

$$\mathbf{z}_{\text{final}} = \text{Dropout}\left(\text{LayerNorm}\left(T^{-1}\sum_{t=1}^{T}\text{MHA}(\mathbf{z}_{\text{se}})_t\right), p = 0.3\right) \in \mathbb{R}^{B\times192}$$

with 4 attention heads (dim=64). The attention output undergoes layer normalization for stabilization and dropout for regularization, producing refined features $\mathbf{z}_{\text{final}}$ for the final classification head. The parallel pathway design enables MSTN to achieve comprehensive temporal modeling where $\mathcal{F}_{\text{CNN}}$ captures local patterns and $\mathcal{F}_{\text{Transformer}}$ models global dependencies, with hierarchical fusion creating synergistic representations that surpass individual pathway capabilities. This architectural innovation establishes a new paradigm for multi-scale time series analysis.

4.1.3. Method-Level Innovation

Many recent time-series models including PatchTST [5] (patch-based tokenization), TimesNet [3] (fixed frequency transformations), and LLM-based approaches like LLM4TS [15] and TIME-LLM [16] (frozen backbone architectures) often rely on rigid, fixed-scale structural priors. This architectural inflexibility forces a fundamental trade-off: models either compromise on capturing multi-scale temporal diversity or suffer from high inference latency, rendering them suboptimal for real-time, resource-constrained deployment.

The proposed MSTN addresses this core challenge through a novel multi-scale hierarchical architecture designed to capture both local patterns and long-range temporal dependencies without structural compromises. The architecture integrates three key components: (i) a multi-scale convolutional encoder that constructs a hierarchical feature pyramid; (ii) a sequence temporal modeling core, which we empirically validate with both BiLSTM and Transformer variants, establishing a flexible foundation for future architectural advancements; and (iii) a gated fusion mechanism augmented with SE and MHTA for dynamic, context-aware feature integration. This hierarchical design enables seamless adaptation to diverse temporal modeling paradigms while maintaining architectural consistency. Our comprehensive experiments demonstrate that these design choices allow MSTN to achieve improved performance compared to SOTA baselines while supporting low-latency inference critical for edge deployment.

4.2. Dataset and Baselines

4.2.1. Long Term Forecasting

The proposed MSTN model was evaluated on eight widely-used benchmark datasets: Weather, Traffic, Electricity, ILI, and four variants of ETT (ETTh1, ETTh2, ETTm1, ETTm2) [1, 3, 5]. These datasets are standard in the field and have been extensively used in prior studies, including EMTSF [14], LLM4TS [15], HiMTM [2], TIME-LLM [16], MTST [13], SOFTS [18] and iTransformer [19]. Notably, large-scale datasets such as Weather, Traffic, and Electricity contain a significantly higher number of time series, making their results more stable and less prone to overfitting compared to smaller datasets. We evaluate 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.

4.2.2. Imputation

The performance of imputation is assessed on benchmark datasets from the ETT [11], Electricity [22], and Weather [23] domains, which commonly encounter missing data in real-world applications. Following the protocol established by TimesNet [3], we evaluated performance under random masking ratios of $12.5\%, 25\%, 37.5\%, 50\%$. The evaluation includes recent SOTA baselines such as GPT2(3) [14], TimesNet [3], PatchTST [5], LightTS [20], and DLinear [6].

4.2.3. Classification

For classification tasks, we utilized ten multivariate benchmark datasets from the UEA time series classification archive [35]. These datasets encompass diverse applications including gesture and action recognition, heartbeat-based medical diagnosis, and other real-world scenarios. The evaluation compares with recent SOTA baselines such as GPT2(6) [14], TimesNet [3], DLinear [6], ETSFormer [36], FEDformer [12], and Informer [11].

4.2.4. SOTA Baselines for Generalizability Study

To rigorously evaluate the generalizability of the proposed MSTN framework, we conducted cross-domain benchmarking across seven publicly available international datasets. The evaluation encompasses datasets from diverse domains: human safety, with the Boubézoul et al. [25] (France) fall event dataset and the Rodegast et al. [21] (Germany) simulator for risky riding behaviour and collisions; general human activity, using UCI-HAR [27] (Italy) and PAMAP2 [29] (USA) to assess robustness in physical activity recognition; animal behavior monitoring via the ActBe-Calf dataset [31] (Ireland) for welfare assessment; and mechanical systems, including MetroPT-3 [32] (Portugal) for predictive maintenance and the NASA Turbofan Engine [34] (USA) for sensor degradation forecasting. This selection tests model transferability across healthcare, activity recognition, agricultural technology, and industrial monitoring. All models were evaluated under a consistent five-fold cross-validation framework to ensure a robust and fair comparison, including recent architectures such as TimesNet [3] and PatchTST [5].

4.3. Training and Model Configuration

The proposed MSTN model is implemented in Python 3.13.1 using PyTorch 2.7.1+cu118 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 pathway (128→64 filters). Training uses AdamW [37] optimizer with learning rate 3×10^{-4}, batch size 64, and task-specific objectives (Focal Loss for classification, MSE for forecasting and imputation). Models train for up to 100 epochs with early stopping based on validation performance for task. Dataset-specific configurations include 96-step lookback for Traffic data and 50-step sequences for physical therapy datasets, with min-max feature normalization applied throughout.

5. Acknowledgments

This research is supported by the IIT Indore Young Faculty Research Catalysing Grant (YFRCG) Scheme (Project ID: IITI/YFRCG/2023-24/01), Indian Institute of Technology Indore, India.

6. Data Availability

All datasets used in this study are publicly available from their respective sources as cited throughout the paper.

7. Author Contributions

Sumit S Shevtekar led the research design, developed the MSTN architecture, implemented all software, conducted experiments across classification, forecasting and imputation tasks, performed data analysis and visualization, and wrote and edited the original draft. **Chandresh K Maurya** contributed to methodology design, experimental validation, and manuscript writing, review and editing. **Gourab Sil** participated in manuscript review and editing. All authors reviewed and approved the final manuscript.

8. Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting (2022). arXiv:2106.13008.
URL https://arxiv.org/abs/2106.13008

[2] S. Zhao, M. Jin, Z. Hou, C. Yang, Z. Li, Q. Wen, Y. Wang, Himtm: Hierarchical multi-scale masked time series modeling with self-distillation for long-term forecasting (2024). arXiv:2401.05012.
URL https://arxiv.org/abs/2401.05012

[3] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, M. Long, Timesnet: Temporal 2d-variation modeling for general time series analysis (2023). arXiv:2210.02186.
URL https://arxiv.org/abs/2210.02186

[4] B. Lim, S. Zohren, Time-series forecasting with deep learning: a survey, Philosophical Transactions of the Royal Society A 379 (2194) (2021) 20200209. doi:10.1098/rsta.2020.0209.

[5] Y. Nie, N. H. Nguyen, P. Sinthong, J. Kalagnanam, A time series is worth 64 words: Long-term forecasting with transformers (2023). arXiv:2211.14730.
URL https://arxiv.org/abs/2211.14730

[6] A. Zeng, M. Chen, L. Zhang, Q. Xu, Are transformers effective for time series forecasting? (2022). arXiv:2205.13504.
URL https://arxiv.org/abs/2205.13504

[7] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.

[8] G. Lai, W.-C. Chang, Y. Yang, H. Liu, Modeling long- and short-term temporal patterns with deep neural networks (2018). arXiv:1703.07015.
URL https://arxiv.org/abs/1703.07015

[9] Y. He, J. Zhao, Temporal convolutional networks for anomaly detection in time series, Journal of Physics: Conference Series 1213 (4) (2019) 042050. doi:10.1088/1742-6596/1213/4/042050.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need (2023). arXiv:1706.03762.
URL https://arxiv.org/abs/1706.03762

[11] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, Proceedings of the AAAI Conference on Artificial Intelligence 35 (12) (2021) 11106–11115.

[12] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin, Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting (2022). arXiv:2201.12740.
URL https://arxiv.org/abs/2201.12740

[13] Y. Zhang, L. Ma, S. Pal, Y. Zhang, M. Coates, Multi-resolution time-series transformer for long-term forecasting (2024). arXiv:2311.04147.
URL https://arxiv.org/abs/2311.04147

[14] M. Alharthi, K. Mahmood, S. Patel, A. Mahmood, Emtsf:extraordinary mixture of sota models for time series forecasting (2025). arXiv:2510.23396.
URL https://arxiv.org/abs/2510.23396

[15] C. Chang, W.-Y. Wang, W.-C. Peng, T.-F. Chen, Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters, ACM Trans. Intell. Syst. Technol. 16 (3) (Apr. 2025). doi:10.1145/3719207.
URL https://doi.org/10.1145/3719207

[16] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, Q. Wen, Time-llm: Time series forecasting by reprogramming large language models (2024). arXiv:2310.01728.
URL https://arxiv.org/abs/2310.01728

[17] T. Zhou, P. Niu, X. Wang, L. Sun, R. Jin, One fits all:power general time series analysis by pretrained lm (2023). arXiv:2302.11939.
URL https://arxiv.org/abs/2302.11939

[18] L. Han, X.-Y. Chen, H.-J. Ye, D.-C. Zhan, Softs: Efficient multivariate time series forecasting with series-core fusion (2024). arXiv:2404.14197.
URL https://arxiv.org/abs/2404.14197

[19] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, M. Long, itransformer: Inverted transformers are effective for time series forecasting (2024). arXiv:2310.06625.
URL https://arxiv.org/abs/2310.06625

[20] T. Zhang, Y. Zhang, W. Cao, J. Bian, X. Yi, S. Zheng, J. Li, Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures (2022). arXiv:2207.01186.
URL https://arxiv.org/abs/2207.01186

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

[22] A. Trindade, ElectricityLoadDiagrams20112014, UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C58C86 (2015).

[23] O. Köllé, Wetterstation. weather., Technical report and dataset, Max-Planck-Institut für Biogeochemie (BGC Jena), Germany, data freely available at https://www.bgc-jena.mpg.de/wetter/ (2025).
URL https://www.bgc-jena.mpg.de/wetter/

[24] P. Rodegast, S. Maier, J. Kneifl, J. Fehr, On using machine learning algorithms for motorcycle collision detection, Discover Applied Sciences 6 (6) (2024) 326.

[25] A. Boubezoul, F. Dufour, S. Bouaziz, S. Espié, Corrigendum to “dataset on powered two wheelers fall and critical events detection”, Data in Brief 30 (2020) 105577. doi:https://doi.org/10.1016/j.dib.2020.105577. URL https://www.sciencedirect.com/science/article/pii/S2352340920304716

[26] F. Elwy, R. Aburukba, A. R. Al-Ali, A. A. Nabulsi, A. Tarek, A. Ayub, M. Elsayeh, Data-driven safe deliveries: The synergy of iot and machine learning in shared mobility, Future Internet 15 (10) (2023).

[27] J. Reyes-Ortiz, D. Anguita, A. Ghio, L. Oneto, X. Parra, Human Activity Recognition Using Smartphones, UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C54S4K (2013).

[28] D. P. Ismi, S. Panchoo, M. Murinto, K-means clustering based filter feature selection on high dimensional data, International Journal of Advances in Intelligent Informatics 2 (2016) 38–45.
URL https://api.semanticscholar.org/CorpusID:43897444

[29] A. Reiss, PAMAP2 Physical Activity Monitoring, UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C5NW2H (2012).

[30] A. Reiss, D. Stricker, Introducing a new benchmarked dataset for activity monitoring, in: 2012 16th International Symposium on Wearable Computers, 2012, pp. 108–109. doi:10.1109/ISWC.2012.13.

[31] O. I. Dissanayake, S. E. McPherson, J. Allyndrée, E. Kennedy, P. Cunningham, L. Riaboff, Actbecalf: Accelerometer-based multivariate time-series dataset for calf behavior classification, Data in Brief 60 (2025) 111462. doi:https://doi.org/10.1016/j.dib.2025.111462.
URL https://www.sciencedirect.com/science/article/pii/S2352340925001945

[32] N. Davari, B. Veloso, R. Ribeiro, J. Gama, MetroPT-3 Dataset, UCI Machine Learning Repository, dOI: https://doi.org/10.24432/C5VW3R (2021).

[33] N. Davari, B. Veloso, R. P. Ribeiro, P. M. Pereira, J. 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), 2021, pp. 1–10. doi:10.1109/DSAA53316.2021.9564181.

[34] A. Saxena, K. Goebel, Nasa turbofan engine degradation simulation data set, nASA Ames Prognostics Center of Excellence (2008).
URL https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/

[35] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, E. Keogh, The uea multivariate time series classification archive, 2018 (2018). arXiv:1811.00075.
URL https://arxiv.org/abs/1811.00075

[36] G. Woo, C. Liu, D. Sahoo, A. Kumar, S. Hoi, Etsformer: Exponential smoothing transformers for time-series forecasting (2022). arXiv:2202.01381.
URL https://arxiv.org/abs/2202.01381

[37] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization (2017). arXiv:1412.6980.