An Enhanced Transformer Encoder Using Inverted Stationarization Decomposition for Long-Term Multivariate Time Series Forecasting

Nasra Pratama Putra
Department of Information Systems, Universitas Musamus, Indonesia | Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia
nasrapratama@unmus.ac.id

Suprapto
Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia
sprapto@ugm.ac.id (corresponding author)

Agus Sihabuddin
Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia
a_sihabudin@ugm.ac.id

Received: 27 November 2025 | Revised: 23 December 2025, 12 January 2026, 21 January 2026, and 26 January 2026 | Accepted: 28 January 2026

 DOI: https://doi.org/10.48084/etasr.16535

ABSTRACT

Autoformer is the first Transformer-based model employing auto-correlation and moving average decomposition to extract trend and seasonal patterns from time series data. Although effective in modeling complex temporal dependencies, these mechanisms show limited accuracy when applied to real-world multivariate time series that are highly non-stationary and lack clear periodicity. This limitation arises from the sensitivity of moving average-based decomposition to fluctuations and distribution shifts, as well as the inability of auto-correlation to capture inter-variable dependencies. To overcome these challenges, this study proposes an Inverted Stationarization Decomposition approach integrated with a Transformer encoder for long-term multivariate time series forecasting. The stationarization process reduces non-stationarity in the input data and restores appropriate statistical properties at the output stage, whereas the inverted embedding mechanism enables effective modeling of correlations among variables. The proposed approach was evaluated using the ETT dataset (four subsets), and the Exchange, Weather, ECL, and Traffic datasets. Experimental results demonstrate that the proposed model consistently outperforms Autoformer, achieving an average Mean Squared Error (MSE) reduction of 29.15% and an average Mean Absolute Error (MAE) reduction of 23.82% across all evaluated datasets.

Keywords-Autoformer; moving average; non-stationary; embedding; forecasting

I. INTRODUCTION

Multivariate time series forecasting aims to predict future sequences for multiple variables over horizons that may significantly exceed the length of the observed data. This task has gained considerable attention due to its wide applicability in domains such as finance, energy systems, weather prediction, and healthcare [1-4]. Traditional Machine Learning (ML) approaches, including Support Vector Machines (SVM) and Adaptive Boosting (AdaBoost), have been applied to this problem; however, these methods often exhibit limited generalization capability and suboptimal forecasting accuracy when handling complex temporal dependencies [5]. To address these limitations, Deep Learning (DL) techniques, such as Recurrent Neural Networks (RNN) [6], Long Short-Term Memory (LSTM) networks [7], and Transformer-based models [8], have been widely adopted due to their superior ability to model nonlinear patterns in time series data.

Among DL-based approaches, Transformer-based models have demonstrated remarkable success in time series forecasting, primarily due to their multi-head attention mechanism, which effectively captures long-range temporal dependencies [9]. Early research efforts in this direction

focused on reducing the quadratic complexity of the self-attention mechanism. By introducing sparse attention structures, models such as LogTrans [10], Reformer [11], and Informer [12] successfully reduced the computational complexity from $O(L^2)$ to $O(L\log L)$. However, such sparsification constraints may lead to the loss of critical temporal information. In complex time series, the accurate identification of latent components, including trends, cycles, and stochastic fluctuations, is essential for improving forecasting performance [13]. Consequently, integrating time series decomposition techniques with advanced predictive models has emerged as an effective strategy for enhancing forecasting accuracy in complex datasets [14]. Following this line of research, decomposition-based Transformer architectures have been proposed to explicitly separate different temporal components of time series data. The Autoformer model employs moving average operations to decompose time series into trend and seasonal components, while maintaining an $O(L\log L)$ computational complexity through an auto-correlation mechanism [15]. Similarly, FedFormer [16] and Inparformer [17] extend this approach by shifting attention computation from the time domain to the frequency domain, while still relying on moving average-based decomposition. In this framework, the moving average estimates the linear trend component [18], and removing the trend yields a more stable seasonal series that is easier to model. Auto-correlation further facilitates the capture of repetitive seasonal patterns. Nevertheless, for non-stationary multivariate time series forecasting tasks with weak or absent periodicity, auto-correlation becomes less effective, and moving average-based decomposition is highly sensitive to distribution shifts [19]. These limitations substantially degrade forecasting performance in real-world scenarios.

To overcome the aforementioned limitations, several robust modeling and decomposition strategies have recently been introduced. Approaches such as Robformer [20] and the Channel Isolation (CI) strategy in PatchTST [21] demonstrate notable improvements in forecasting accuracy by enhancing robustness to non-stationarity. In parallel, attention mechanisms specifically designed for non-stationary data, including de-stationary attention in the Non-stationary Transformer [22] and two-stage attention in Crossformer [23], have shown competitive performance in multivariate forecasting tasks. Despite these advances, most existing models largely overlook inter-variable dependencies, which are critical when modeling complex real-world multivariate time series. Conversely, recent studies indicate that combining moving average-based decomposition with direct linear modeling can achieve comparable forecasting accuracy under certain conditions [24], highlighting the need for more principled approaches that jointly address non-stationarity and variable-wise dependencies.

Motivated by these observations, this study aims to develop a more accurate and computationally efficient forecasting framework by integrating Autoformer-style decomposition with data stationarization and an inverted embedding mechanism, referred to as Inverted Stationarization Decomposition. The proposed stationarization process employs instance normalization to alleviate non-stationarity in the input data, whereas reverse normalization restores the statistical properties at the output stage. Furthermore, the inverted embedding mechanism enables the encoder to better capture inter-variable correlations by applying attention across variable dimensions rather than being confined to temporal positions. To ensure computational efficiency, the proposed model adopts a lightweight architecture that relies solely on encoder layers and a single decomposition process, without modifying the core structure of the Vanilla Transformer.

II. METHODOLOGY

This study proposes an Inverted Stationarization Decomposition framework to enhance the long-term multivariate time series forecasting accuracy of a Transformer encoder. A multivariate time series is defined as a sequence of historical observations $X = [x_1, x_2, \dots, x_L] \in \mathbb{R}^{L\times C}$, where L denotes the number of time steps and C represents the number of variables. The objective is to predict future values over the next S time steps, denoted as $Y = [x_{L+1}, x_{L+2}, \dots, x_{L+S}] \in \mathbb{R}^{S\times C}$. The parameters L and S correspond to the look-back window and the prediction horizon, respectively. The overall architecture of the proposed model is illustrated in Figure 1.

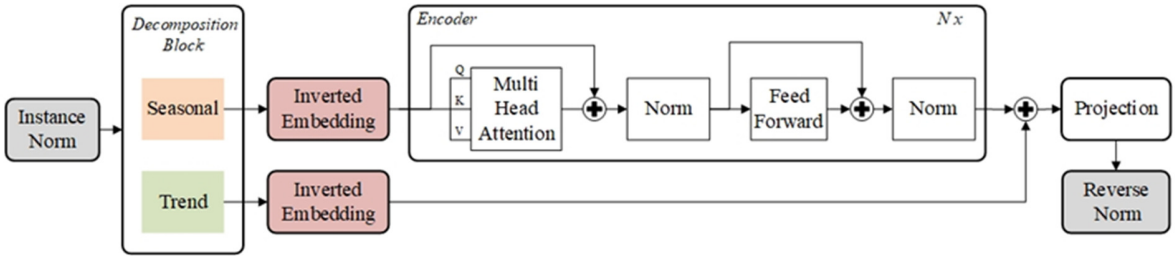

Fig. 1. Architectural model of the proposed method.

A. Stationarization Block

During inference, non-stationary time series often induce distribution shifts between the training and testing data, primarily due to variations in mean and standard deviation. Reversible Instance Normalization (RevIN) addresses this issue by introducing learnable affine parameters for each input series, applying instance normalization and subsequently restoring the original statistics at the output stage [25], thereby enforcing distributional consistency across instances. In contrast, the proposed stationarization block adopts a non-learnable instance normalization strategy that preserves the reversible normalization property while eliminating additional

trainable parameters. The instance normalization and reverse normalization components jointly constitute the stationarization block, the detailed formulation of which is presented as follows.

1) Instance Normalization Module

This module transforms each input series X through translation and scaling operations using (1), (2), and (3):

$$\mu_x = \frac{1}{L}\sum_{i=1}^{L} x_i \tag{1}$$

$$\sigma_x = \sqrt{\frac{1}{L}\sum_{i=1}^{L}(x_i - \mu_x)^2} \tag{2}$$

$$x_i' = \frac{x_i}{\sigma_x} \tag{3}$$

where μ_x is the mean value, σ_x is the standard deviation, x_i' denotes the normalized value, and $\mu_x, \sigma_x \in \mathbb{R}^{C\times 1}$.

2) Reverse Normalization Module

After the model predicts future series values $Y' = [y_1', y_2', \dots, y_S'] \in \mathbb{R}^{S\times C}$, this module restores the original scale using the saved mean μ_x and standard deviation σ_x using (4):

$$\hat{y}_i = \sigma_x \odot (y_i' + \mu_x) \tag{4}$$

where y_i' is projected output value, $\hat{y}_i$ denotes the final prediction, and $\odot$ represents element-wise multiplication. This reversible mechanism effectively mitigates distribution shifts, enhancing model stability.

B. Decomposition Block

The decomposition block analyzes the time series by explicitly separating it into a trend component and a seasonal component, following the Autoformer framework. Moving average operations are employed as a fundamental technique to estimate the trend component, from which the seasonal component is subsequently derived [26]. For the normalized input series $X' = [x_1', x_2', \dots, x_L'] \in \mathbb{R}^{L\times C}$, the moving average-based decomposition is formulated as follows:

$$X_t' = \frac{x_1' + \cdots + x_L'}{L} = \mathrm{AvgPool}\big(\mathrm{Padding}(X')\big) \tag{5}$$

$$X_s' = X' - X_t' \tag{6}$$

where X_s' denotes the seasonal features, X_t' denotes the trend features, and both $X_s', X_t' \in \mathbb{R}^{L\times C}$. The $\mathrm{AvgPool}(\cdot)$ acts as a moving average, with $\mathrm{Padding}$ applied to maintain the original series length.

C. Inverted Embedding Block

In the Vanilla Transformer architecture, variables observed at the same timestamp are aggregated into a single token, which requires the model to implicitly learn inter-variable relationships within a local temporal context. Inspired by the DLinear model, which employs independent linear transformations by treating each variable as an individual token across the entire time series, this study adopts an inverted embedding strategy. Specifically, inverted embedding is applied to transform the seasonal and trend feature matrices, $X_s', X_t' \in \mathbb{R}^{L\times C}$, into variable-wise representations X_{is}' and $X_{it}' \in \mathbb{R}^{C\times D}$, as defined in (7) and (8):

$$X_{is}' = \mathrm{Linear}(X_s'.\mathrm{transpose}) \tag{7}$$

$$X_{it}' = \mathrm{Linear}(X_t'.\mathrm{transpose}) \tag{8}$$

where X_{is}' and X_{it}' are the embedded seasonal and trend features, respectively. The Linear is implemented via a Multi-Layer Perceptron (MLP) network, and the $\mathrm{transpose}$ operator transposes the feature dimensions.

D. Encoder Block

The encoder block is employed to evaluate the performance gains achieved by utilizing decomposed input representations. In general, the encoder comprises three main components: a self-attention module, a layer normalization module, and a Feed-Forward Network (FFN) module. The detailed formulation of each component is described as follows.

1) Self-Attention Module

The self-attention mechanism enables the model to assess the relative importance of different tokens within the input sequence, thereby facilitating the extraction of long-range dependencies and contextual relationships. Specifically, the query (Q), key (K), and value (V) matrices are obtained through linear projections of the input representations, as defined in (9):

$$Q = XW_Q,\ K = XW_K,\ V = XW_V \tag{9}$$

where W_Q, W_K, W_V are the learned weight matrices. Using multi-head attention, a number of queries (Q), keys (K), and values (V) are fed into a multi-scale dot-product attention block using (10):

$$\mathrm{Attention} = \mathrm{Softmax}\left(\frac{QK^{\mathrm{Transpose}}}{\sqrt{d_k}}\right) \times V \tag{10}$$

where $\sqrt{d_k}$ is the scaling factor and the $\mathrm{Softmax}$ function produces a probability distribution. The computational complexity for attention scores is $O(n^2 d)$.

2) Layer Normalization Module

Layer normalization is applied to improve training stability and convergence in deep Transformer architectures. The layer normalization operation in the inverted encoder is defined in (11):

$$\mathrm{LayerNorm}(H) = \frac{h_n - \mathrm{Mean}(h_n)}{\sqrt{\mathrm{Var}(h_n)}} \tag{11}$$

where $h_n \in \mathbb{R}^{C\times D}$ denotes the output of the self-attention module for the n-th variable, and $n = 1,2,3, \dots, C$. The Mean and the Var functions are used for normalization.

3) Feed-Forward Network Module

In Transformer architectures, the FFN is employed to encode token representations through position-wise nonlinear transformations that are applied identically to each token. The FFN operation in the inverted encoder is defined in (12):

$$\mathrm{FNN}(H) = \mathrm{GeLU}(HW^1 + b^1)W^2 + b^2 \tag{12}$$

where H is the output of the previous normalization module, and W^1, b^1, W^2, b^2 are learnable parameters. The GeLU

activation function introduces non-linearity, improves gradient flow, and accelerates convergence during training.

III. EXPERIMENTAL RESULTS

A. Data and Experiment Setting

Experimental analyses were conducted on eight real-world datasets: ETT (four subsets), Exchange, Weather, ECL, and Traffic, which are commonly used for evaluating Autoformer and other comparative models. These datasets are publicly available in the Autoformer code repository [27]. Table I presents the overall statistics of the datasets.

TABLE I. DETAILED DATASET DESCRIPTIONS

Dataset	Dim	Size (train, validation, test)
ETTm1	7	(34465, 11521, 11521)
ETTm2	7	(34465, 11521, 11521)
ETTh1	7	(8545, 2881, 2881)
ETTh2	7	(8545, 2881, 2881)
Exchange	8	(5120, 665, 1422)
Weather	21	(36792, 5271, 10540)
ECL	321	(18317, 2633, 5261)
Traffic	862	(12185, 1757, 3509)

Dim denotes the number of variates in each dataset. The ETT dataset, collected between July 2016 and July 2018, contains seven factors related to electricity transformers, with four subsets: ETTh1, ETTh2, ETTm1, and ETTm2. The Exchange dataset comprises daily exchange rate data from eight countries (1990–2016). The Weather dataset includes 21 meteorological parameters recorded every 10 min at the Max Planck Biogeochemistry Institute's weather station in 2020. The ECL dataset records hourly electricity consumption from 321 clients. The Traffic dataset consists of hourly road occupancy rates from 862 sensors in the San Francisco Bay Area (2015–2016). Following established practice, datasets are split into training, validation, and test sets with ratios of 6:2:2 for ETT and 7:1:2 for the other datasets. The proposed Inverted Stationarization Decomposition approach is implemented using the PyTorch framework and evaluated on a single NVIDIA Tesla T4 GPU. Model optimization is performed using the Adam optimizer with initial learning rates selected from {0.001, 0.0005, 0.0001}, whereas the L2 loss function is employed as the training objective. The batch size is fixed at 32, and the maximum number of training epochs is set to 10. The number of encoder block E is selected from {2, 3, 4}, and the dimension of series representations D is chosen from {256, 512}.

B. Model Evaluation

For all long-term forecasting tasks, Mean Absolute Error (MAE) and Mean Squared Error (MSE) are used to compare model performance. MAE and MSE are determined using (13) and (14):

$$\text{MSE} = \frac{1}{L}\sum_{i=1}^{L}(y_i - \hat{y}_i)^2 \quad (13)$$

$$\text{MAE} = \frac{1}{L}\sum_{i=1}^{L}|(y_i - \hat{y}_i)| \quad (14)$$

where y_i denotes the actual value, $\hat{y}_i$ denotes the predicted value, and L represents the number of time steps.

C. Main Forecasting Results

In this section, the performance of the proposed approach is systematically evaluated and compared with existing forecasting methods. Six representative benchmark models are selected for comparison, including (1) Transformer-based models: Informer, Autoformer, FEDformer, Non-stationary, and Crossformer; and (2) linear-based model: Dlinear. Table II reports the comparative results for long-term multivariate time series forecasting under prediction horizons $S \in \{96, 192, 336, 720\}$ with a fixed look-back window of $L = 96$. Forecasting accuracy is evaluated using MSE and MAE, where lower values indicate better performance, and the best results are highlighted in bold. Overall, the proposed method achieves superior performance across most experimental configurations, demonstrating strong predictive capability. Notable performance improvements are observed on multiple datasets, including ETTm1 (21.6%), ETTm2 (62.7%), ETTh1 (13.2%), ETTh2 (59.7%), ECL (27.9%), Exchange (62.0%), and Traffic (22.9%) in terms of average MSE reduction, whereas relatively smaller gains are observed on the Weather dataset when compared to recent benchmark models.

Several benchmark models are capable of achieving lower average MSE values than the proposed approach under specific settings. Experimental results on the ETTh1 and Exchange datasets indicate that the Dlinear model attains marginally better average MSE, with improvements of 0.7% and 0.8%, respectively, compared to the proposed method. Similarly, on the Weather dataset, the Crossformer model slightly outperforms the proposed approach, achieving a 0.1% improvement in forecasting accuracy. Nevertheless, when the comparison is restricted to the Autoformer model, the proposed approach demonstrates substantial performance gains, achieving an average MSE reduction of 29.15% and an average MAE reduction of 23.82% across all evaluated benchmarks.

IV. DISCUSSION

This section provides a comprehensive discussion of the experimental results through multiple evaluation perspectives. Specifically, ablation studies are first conducted to examine the contribution of each model component, followed by paired t-tests to assess the statistical significance of performance differences. In addition, training dynamics are analyzed to identify potential overfitting or underfitting behavior, and qualitative visualizations are presented to further illustrate the forecasting performance of the proposed approach. Table III presents the ablation study results on the ETT datasets under multiple prediction horizons $S \in \{96, 192, 336, 720\}$. Removing the trend component ("w/o trend") leads to noticeable performance degradation, highlighting the importance of explicitly modeling long-term temporal patterns, whereas the impact of excluding seasonal–trend interactions is relatively limited in certain datasets. The absence of the inverted embedding mechanism ("w/o inverted embedding") results in increased forecasting errors, particularly on the ETTh datasets and for longer horizons, underscoring the role of variate-wise tokenization in capturing inter-variable dependencies. Most notably, eliminating the stationarization block ("w/o stationarization") causes severe performance deterioration across all datasets, indicating its critical role in

mitigating non-stationarity and distribution shifts. Overall, the proposed model achieves superior MSE performance in 13 out of 16 experimental cases across all prediction horizons.

TABLE II. PERFORMANCE COMPARISON OF THE PROPOSED METHOD WITH BENCHMARK MODELS

Model		Proposed method		Crossformer		Dlinear		FEDformer		Non-stationary		Autoformer		Informer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.343	0.371	0.404	0.426	0.345	0.372	0.379	0.419	0.386	0.398	0.505	0.475	0.611	0.551
	192	0.378	0.392	0.450	0.451	0.380	0.389	0.426	0.441	0.459	0.444	0.553	0.496	0.671	0.582
	336	0.410	0.414	0.532	0.515	0.413	0.413	0.445	0.459	0.495	0.464	0.621	0.537	0.846	0.682
	720	0.475	0.448	0.666	0.589	0.474	0.453	0.543	0.490	0.585	0.516	0.671	0.561	0.943	0.732
	Avg	0.402	0.406	0.513	0.495	0.403	0.407	0.448	0.452	0.481	0.456	0.588	0.517	0.768	0.637
ETTm2	96	0.178	0.262	0.287	0.366	0.193	0.292	0.203	0.287	0.192	0.274	0.255	0.339	0.365	0.453
	192	0.243	0.305	0.414	0.492	0.284	0.362	0.269	0.328	0.280	0.339	0.281	0.340	0.533	0.563
	336	0.301	0.343	0.597	0.542	0.369	0.427	0.325	0.366	0.334	0.361	0.339	0.372	1.363	0.887
	720	0.404	0.399	1.730	1.042	0.554	0.522	0.421	0.415	0.417	0.413	0.433	0.432	3.379	1.388
	Avg	0.282	0.327	0.757	0.611	0.350	0.401	0.305	0.349	0.306	0.347	0.327	0.371	1.410	0.823
ETTh1	96	0.387	0.402	0.423	0.448	0.386	0.400	0.376	0.419	0.513	0.491	0.449	0.459	0.806	0.648
	192	0.445	0.436	0.471	0.474	0.437	0.432	0.420	0.448	0.534	0.504	0.500	0.482	0.942	0.730
	336	0.500	0.466	0.570	0.546	0.481	0.459	0.459	0.465	0.588	0.535	0.521	0.496	1.139	0.845
	720	0.504	0.485	0.653	0.621	0.519	0.516	0.506	0.507	0.643	0.616	0.514	0.512	1.200	0.862
	Avg	0.459	0.447	0.529	0.522	0.456	0.452	0.440	0.460	0.570	0.537	0.496	0.487	1.022	0.771
ETTh2	96	0.295	0.345	0.745	0.584	0.333	0.387	0.358	0.397	0.476	0.458	0.346	0.388	2.488	1.310
	192	0.379	0.396	0.877	0.656	0.477	0.476	0.429	0.439	0.512	0.493	0.456	0.452	4.318	1.768
	336	0.419	0.430	1.043	0.731	0.594	0.541	0.496	0.487	0.552	0.551	0.482	0.486	4.217	1.788
	720	0.427	0.445	1.104	0.763	0.831	0.657	0.463	0.474	0.562	0.560	0.515	0.511	4.288	1.798
	Avg	0.380	0.404	0.942	0.684	0.559	0.515	0.437	0.449	0.526	0.516	0.450	0.459	3.828	1.666
ECL	96	0.148	0.241	0.219	0.314	0.197	0.282	0.193	0.308	0.169	0.273	0.201	0.317	0.274	0.368
	192	0.162	0.253	0.231	0.322	0.196	0.285	0.201	0.315	0.182	0.286	0.222	0.334	0.296	0.386
	336	0.179	0.273	0.246	0.337	0.209	0.301	0.214	0.329	0.200	0.304	0.231	0.338	0.300	0.394
	720	0.213	0.303	0.280	0.363	0.245	0.333	0.246	0.355	0.222	0.321	0.254	0.361	0.373	0.439
	Avg	0.176	0.268	0.244	0.334	0.212	0.300	0.214	0.327	0.193	0.296	0.227	0.338	0.311	0.397
Exchange	96	0.083	0.203	0.256	0.367	0.088	0.218	0.148	0.278	0.111	0.237	0.197	0.323	0.847	0.752
	192	0.174	0.298	0.470	0.509	0.176	0.315	0.271	0.315	0.219	0.335	0.300	0.369	1.204	0.895
	336	0.329	0.415	1.268	0.883	0.313	0.427	0.460	0.427	0.421	0.476	0.509	0.524	1.672	1.036
	720	0.842	0.691	1.767	1.068	0.839	0.695	1.195	0.695	1.092	0.769	1.447	0.941	2.478	1.310
	Avg	0.357	0.402	0.940	0.707	0.354	0.414	0.519	0.429	0.461	0.454	0.613	0.539	1.550	0.998
Traffic	96	0.392	0.268	0.522	0.290	0.650	0.396	0.587	0.366	0.612	0.338	0.613	0.388	0.719	0.391
	192	0.412	0.276	0.530	0.293	0.598	0.370	0.604	0.373	0.613	0.340	0.616	0.382	0.696	0.379
	336	0.428	0.284	0.558	0.305	0.605	0.373	0.621	0.383	0.618	0.328	0.622	0.337	0.777	0.420
	720	0.463	0.303	0.589	0.328	0.645	0.394	0.626	0.382	0.653	0.355	0.660	0.408	0.864	0.472
	Avg	0.424	0.283	0.550	0.304	0.625	0.383	0.610	0.376	0.624	0.340	0.628	0.379	0.764	0.416
Weather	96	0.176	0.218	0.158	0.230	0.196	0.255	0.217	0.296	0.173	0.223	0.266	0.336	0.300	0.384
	192	0.224	0.259	0.206	0.277	0.237	0.296	0.276	0.336	0.245	0.285	0.307	0.367	0.598	0.544
	336	0.279	0.299	0.272	0.335	0.283	0.335	0.339	0.380	0.321	0.338	0.359	0.395	0.578	0.523
	720	0.356	0.348	0.398	0.418	0.345	0.381	0.403	0.428	0.414	0.410	0.419	0.428	1.059	0.741
	Avg	0.259	0.281	0.259	0.315	0.265	0.317	0.309	0.360	0.288	0.314	0.338	0.382	0.634	0.548

TABLE III. ABLATION STUDY RESULTS ON THE ETT DATASET

Method	Metric	ETTm1				ETTm2				ETTh1				ETTh2			
		96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
Purposed method	MSE	0.343	0.378	0.410	0.475	0.178	0.243	0.301	0.404	0.387	0.445	0.500	0.504	0.295	0.379	0.419	0.427
	MAE	0.371	0.392	0.414	0.448	0.262	0.305	0.343	0.399	0.402	0.436	0.466	0.485	0.345	0.396	0.430	0.445
W/O trend	MSE	0.497	0.511	0.520	0.557	0.217	0.272	0.323	0.421	0.397	0.451	0.503	0.504	0.327	0.403	0.431	0.436
	MAE	0.467	0.475	0.482	0.501	0.295	0.327	0.359	0.411	0.409	0.441	0.469	0.486	0.366	0.411	0.439	0.450
W/O seasonal-trend	MSE	0.342	0.383	0.426	0.491	0.186	0.254	0.317	0.416	0.387	0.441	0.494	0.508	0.301	0.381	0.428	0.430
	MAE	0.377	0.396	0.420	0.459	0.272	0.314	0.353	0.409	0.405	0.435	0.467	0.493	0.350	0.399	0.432	0.446
W/O inverted embedding	MSE	0.368	0.396	0.442	0.516	0.211	0.328	0.349	0.445	0.509	0.537	0.553	0.632	0.388	0.427	0.507	0.466
	MAE	0.396	0.415	0.445	0.488	0.290	0.367	0.371	0.419	0.485	0.507	0.510	0.572	0.398	0.430	0.490	0.467
W/O stationarization	MSE	0.934	0.819	0.913	0.974	4.109	3.924	5.089	3.953	0.873	0.994	1.001	0.981	4.235	3.720	3.850	3.405
	MAE	0.733	0.695	0.745	0.768	1.504	1.488	1.741	1.472	0.714	0.757	0.760	0.771	1.518	1.417	1.438	1.339

Table IV presents the outcomes of paired two-tailed t-tests comparing the proposed method with multiple benchmark models. All hypothesis tests were conducted at a significance level of $\alpha = 0.05$. For each comparison, the resulting p-values are substantially below the specified significance threshold. Specifically, the p-values across all benchmark comparisons

range from 1.2×10^{-3} to 6.0×10^{-8}, satisfying the condition $p < \alpha$. Consequently, the null hypothesis (H_0) of no difference in mean forecasting performance is rejected in all cases, indicating statistically significant differences between the proposed method and the corresponding benchmark models. Under the experimental conditions considered, these results provide strong inferential evidence supporting the conclusion that the proposed method achieves superior forecasting accuracy relative to the evaluated benchmarks.

In addition to the statistical validation, the training dynamics of the proposed model are examined to assess potential overfitting or underfitting behavior. Specifically, loss curves on the ETTh1 dataset with a prediction horizon of 96 are analyzed and compared with those of Autoformer, as illustrated in Figure 2. The proposed model exhibits stable and consistent convergence, with training and validation losses decreasing in a closely aligned manner. In contrast, Autoformer shows early signs of overfitting, as the training loss continues to decrease while the validation loss begins to increase from the first epoch. Based on these observations, restricting the training process to 10 epochs provides a balanced and fair evaluation across all compared models and helps ensure comparable training conditions.

Finally, qualitative visualization is used to further illustrate the forecasting behavior of the proposed method. As shown in Figure 3, observable discrepancies remain between the predicted and ground-truth values. This is primarily due to the fact that the visualization displays only a single variate (the last feature) from each dataset, which does not fully reflect the multivariate forecasting performance. For instance, in the Weather dataset, which consists of 21 variables, overall MSE values are influenced by all features rather than a single variate. Nevertheless, for the ECL and Traffic datasets, the visualized variate contributes noticeably to the overall error reduction, aligning with the quantitative improvements observed in the evaluation metrics.

TABLE IV. PAIRED T-TESTS COMPARING THE PROPOSED METHOD WITH BENCHMARK MODELS

Pair	Comparison	Mean	Std. deviation	*t*-Statistic	*df*	*t*-critical	*p*-value
1	Proposed method – Crossformer	-0.249625	0.322305	-4.381231	31	2.039513	0.00012534
2	Proposed method – Dlinear	-0.060750	0.096484	-3.561771	31	2.039513	0.00121383
3	Proposed method – FEDformer	-0.067844	0.078415	-4.894246	31	2.039513	0.00002908
4	Proposed method – Non-stationary	-0.088906	0.071066	-7.076960	31	2.039513	0.00000006
5	Proposed method – Autoformer	-0.116125	0.110124	-5.965103	31	2.039513	0.00000136
6	Proposed method - Informer	-0.943625	1.143719	-4.667185	31	2.039513	0.00005563

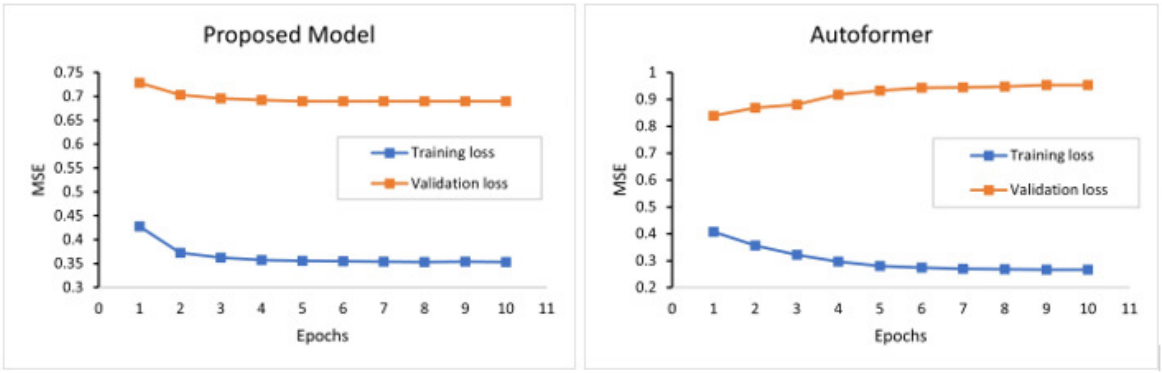

Fig. 2. Training and validation loss curves for the ETTh1 dataset.

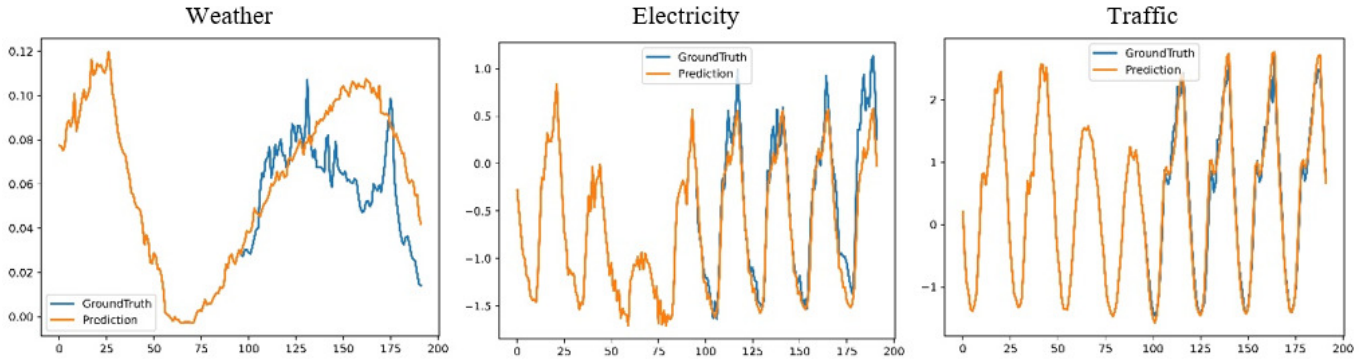

Fig. 3. Visualization of the forecasting results obtained by the proposed method.

V. CONCLUSION

In this paper, an enhanced Transformer encoder incorporating Inverted Stationarization Decomposition is proposed for long-term multivariate time series forecasting. Unlike conventional Transformer-based models that rely on temporal tokenization and often overlook inter-variable dependencies, the proposed approach employs variate-wise tokens generated through inverted embedding. The main contribution of this work lies in demonstrating that, without modifying the self-attention mechanism and by using an encoder-only architecture, the integration of inverted embedding with stationarized decomposition can effectively improve forecasting performance. Extensive experiments on multiple benchmark datasets confirm the effectiveness of the proposed method, achieving average Mean Squared Error (MSE) reductions of 31.65% on ETTm1, 13.91% on ETTm2, 7.45% on ETTh1, 15.50% on ETTh2, 22.68% on ECL, 41.78% on Exchange, 32.49% on Traffic, and 23.39% on Weather when compared with the Autoformer model.

ACKNOWLEDGMENT

This work was supported by the Department of Computer Science and Electronics, Universitas Gadjah Mada.

REFERENCES

[1] S. Martelo, D. León, and G. Hernandez, "Multivariate Financial Time Series Forecasting with Deep Learning," in *9th Workshop on Engineering Applications on Applied Computer Sciences in Engineering*, Bogotá, Colombia, 2022, pp. 160–169, https://doi.org/10.1007/978-3-031-20611-5_14.

[2] R. Pérez-Chacón, G. Asencio-Cortés, A. Troncoso, and F. Martínez-Álvarez, "Pattern sequence-based algorithm for multivariate big data time series forecasting: Application to electricity consumption," *Future Generation Computer Systems*, vol. 154, pp. 397–412, May 2024, https://doi.org/10.1016/j.future.2023.12.021.

[3] H. Zhang, J. Wang, Y. Qian, and Q. Li, "Point and interval wind speed forecasting of multivariate time series based on dual-layer LSTM," *Energy*, vol. 294, May 2024, Art. no. 130875, https://doi.org/10.1016/j.energy.2024.130875.

[4] D. B. N. Assad, J. Cara, and M. Ortega-Mier, "Comparing Short-Term Univariate and Multivariate Time-Series Forecasting Models in Infectious Disease Outbreak," *Bulletin of Mathematical Biology*, vol. 85, no. 1, Dec. 2022, Art. no. 9, https://doi.org/10.1007/s11538-022-01112-5.

[5] Z. Chen, M. Ma, T. Li, H. Wang, and C. Li, "Long sequence time-series forecasting with deep learning: A survey," *Information Fusion*, vol. 97, Sept. 2023, Art. no. 101819, https://doi.org/10.1016/j.inffus.2023.101819.

[6] E. A. Nketiah, L. Chenlong, J. Yingchuan, and S. A. Aram, "Recurrent neural network modeling of multivariate time series and its application in temperature forecasting," *Plos One*, vol. 18, no. 5, May 2023, Art. no. e0285713, https://doi.org/10.1371/journal.pone.0285713.

[7] J. Ju and F.-A. Liu, "Multivariate Time Series Data Prediction Based on ATT-LSTM Network," *Applied Sciences*, vol. 11, no. 20, Oct. 2021, Art. no. 9373, https://doi.org/10.3390/app11209373.

[8] A. Vaswani *et al.*, "Attention is All you Need," in *31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 5998–6008.

[9] Q. Wen *et al.*, "Transformers in Time Series: A Survey," in *Thirty-Second International Joint Conference on Artificial Intelligence*, Macao, China, 2023, pp. 6778–6786, https://doi.org/10.24963/ijcai.2023/759.

[10] S. Li *et al.*, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Vancouver, Canada, 2019, pp. 5243–5253.

[11] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The Efficient Transformer." arXiv, Feb. 18, 2020, https://doi.org/10.48550/arXiv.2001.04451.

[12] H. Zhou *et al.*, "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 11106–11115, May 2021, https://doi.org/10.1609/aaai.v35i12.17325.

[13] H. Han *et al.*, "Time series forecasting model for non-stationary series pattern extraction using deep learning and GARCH modeling," *Journal of Cloud Computing*, vol. 13, no. 1, Jan. 2024, Art. no. 2, https://doi.org/10.1186/s13677-023-00576-7.

[14] N.-H. Duong, M.-T. Nguyen, T.-H. Nguyen, and T.-P. Tran, "Application of Seasonal Trend Decomposition using Loess and Long Short-Term Memory in Peak Load Forecasting Model in Tien Giang," *Engineering, Technology & Applied Science Research*, vol. 13, no. 5, pp. 11628–11634, Oct. 2023, https://doi.org/10.48084/etasr.6181.

[15] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: decomposition transformers with auto-correlation for long-term series forecasting," in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, Virtual Event, 2021, pp. 22419–22430.

[16] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting," in *Proceedings of the 39th International Conference on Machine Learning*, Baltimore, MD, USA, 2022, pp. 27268–27286.

[17] H. Cao, Z. Huang, T. Yao, J. Wang, H. He, and Y. Wang, "InParformer: Evolutionary Decomposition Transformers with Interactive Parallel Attention for Long-Term Time Series Forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, pp. 6906–6915, June 2023, https://doi.org/10.1609/aaai.v37i6.25845.

[18] M. Höll, K. Kiyono, and H. Kantz, "Theoretical foundation of detrending methods for fluctuation analysis such as detrended fluctuation analysis and detrending moving average," *Physical Review E*, vol. 99, no. 3, Mar. 2019, Art. no. 033305, https://doi.org/10.1103/PhysRevE.99.033305.

[19] Y. Wang, J. Zhu, and R. Kang, "DESTformer: A Transformer Based on Explicit Seasonal–Trend Decomposition for Long-Term Series Forecasting," *Applied Sciences*, vol. 13, no. 18, Sept. 2023, Art. no. 10505, https://doi.org/10.3390/app131810505.

[20] Y. Yu, R. Ma, and Z. Ma, "Robformer: A robust decomposition transformer for long-term time series forecasting," *Pattern Recognition*, vol. 153, Sept. 2024, Art. no. 110552, https://doi.org/10.1016/j.patcog.2024.110552.

[21] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A Time Series is Worth 64 Words: Long-term Forecasting with Transformers." arXiv, Mar. 05, 2023, https://doi.org/10.48550/arXiv.2211.14730.

[22] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: exploring the stationarity in time series forecasting," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, New Orleans, LA, USA, 2022, pp. 9881–9893.

[23] Y. Zhang and J. Yan, "Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting," in *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda, 2022.

[24] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are Transformers Effective for Time Series Forecasting?," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, pp. 11121–11128, June 2023, https://doi.org/10.1609/aaai.v37i9.26317.

[25] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift," in *The Tenth International Conference on Learning Representations*, Virtual Event, 2021.

[26] G. Kavakci, B. Cicekdag, and S. Ertekin, "Time Series Prediction of Solar Power Generation Using Trend Decomposition," *Energy Technology*, vol. 12, no. 2, Feb. 2024, Art. no. 2300914, https://doi.org/10.1002/ente.202300914.

[27] "Autoformer." Google Drive. https://drive.google.com/drive/folders/1ZOYpTUa82_jCcxIdTmyr0LXQfvaM9vIy.