

Adaptive Convolutional Forecasting Network Based on Time Series Feature-Driven

Dandan Zhang^a, Zhiqiang Zhang^a, Nanguang Chen^b and Yun Wang^a

^a*School of Computer Science and Engineering, Southeast University, Nanjing, China*

^b*College of Design and Engineering, National University of Singapore, Singapore*

ARTICLE INFO

Keywords:

Nonlinear feature
Deformable convolution
Multi-resolution convolution
Time series forecasting


ABSTRACT

Time series data in real-world scenarios contain a substantial amount of nonlinear information, which significantly interferes with the training process of models, leading to decreased prediction performance. Therefore, during the time series forecasting process, extracting the local and global time series patterns and understanding the potential nonlinear features among different time observations are highly significant. To address this challenge, we introduce multi-resolution convolution and deformable convolution operations. By enlarging the receptive field using convolution kernels with different dilation factors to capture temporal correlation information at different resolutions, and adaptively adjusting the sampling positions through additional offset vectors, we enhance the network's ability to capture potential nonlinear features among time observations. Building upon this, we propose ACNet, an adaptive convolutional network designed to effectively model the local and global temporal dependencies and the nonlinear features between observations in multivariate time series. Specifically, by extracting and fusing time series features at different resolutions, we capture both local contextual information and global patterns in the time series. The designed nonlinear feature adaptive extraction module captures the nonlinear features among different time observations in the time series. We evaluated the performance of ACNet across twelve real-world datasets. The results indicate that ACNet consistently achieves state-of-the-art performance in both short-term and long-term forecasting tasks with favorable runtime efficiency.

1. Introduction

Time-series forecasting (TSF) is one of the most critical challenges in time series analysis, with wide-ranging applications in fields such as energy [1, 2, 3], traffic [4], weather [5], disease [6], and more. Its core objective is to leverage past time series data to forecast changing trends over a future time horizon. Effective feature extraction from time series data is essential for improving forecasting accuracy. However, time series data in real-world scenarios often exhibit strong nonlinear features due to factors such as environmental conditions and equipment usage. This poses significant challenges for effective feature extraction. Firstly, there is an increase in noise and uncertainty: data collected by sensors are highly sensitive to environmental changes and external disturbances, leading to increased noise and uncertainty in the time series. Secondly, the extraction of complex dependencies is difficult: time series data in real-world scenarios may exhibit complex nonlinear and dynamic patterns, and the inherent dependency structures and patterns between time observations cannot be accurately captured and described by simple linear models. Additionally, the distribution of time series data in real-world scenarios changes over time, posing higher demands on local feature extraction to effectively capture short-term dynamic behaviors.

*Corresponding author

 ywang_cse@seu.edu.cn (Y. Wang)

ORCID(s):

To further illustrate the complex nonlinear characteristics of time series data in real-world scenarios, we performed phase space reconstruction on real datasets from different fields. Figure 1 shows the phase space reconstruction diagrams of these datasets, from which the following points can be observed: 1) the trajectory shapes are complex and variable; 2) the trajectories cluster in multiple specific regions; 3) the trajectories are close to each other and intricately intertwined. These characteristics indicate that there are strong nonlinear relationships and dynamic behaviors between different time observations in the time series, which greatly limit the effective extraction of time series data features.

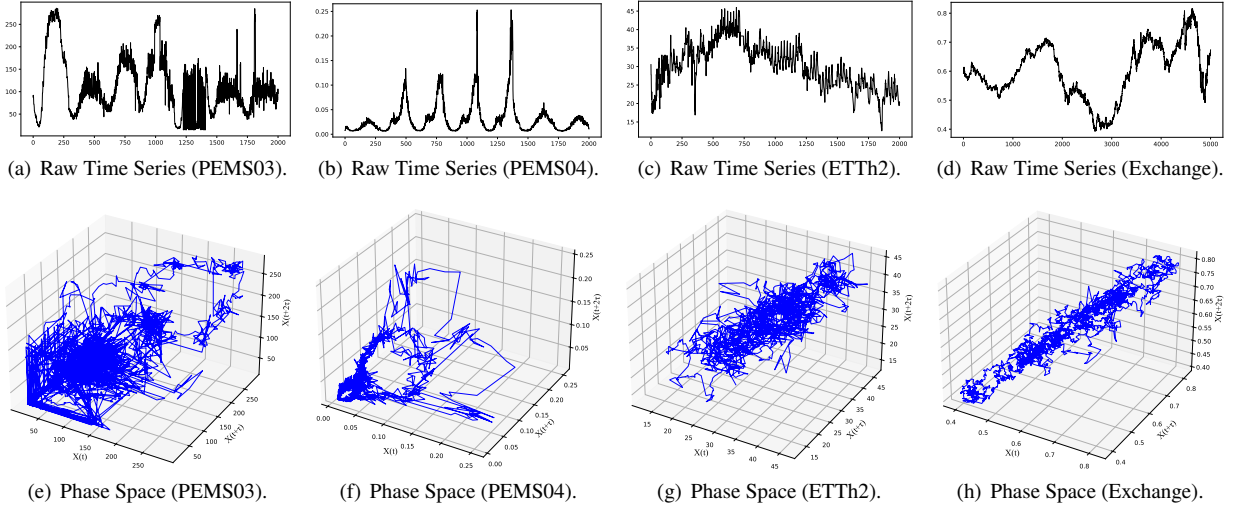


Figure 1: Nonlinear Features of Data.

To effectively extract complex features and patterns in multivariate time series, research focus on multivariate time series forecasting methods has gradually shifted towards Transformer-based models, which employ multi-encoder-decoder architectures and multi-head self-attention mechanisms. This shift is attributed to the strong capabilities of Transformer models in capturing temporal dependencies in long sequences. Consequently, a series of Transformer-based model-driven approaches have emerged, including PatchTST [7], iTransformer [8], and Crossformer [9]. Thanks to the ability of the self-attention mechanism to extract long-term dependencies in time series, transformer-based models perform well in TSF. However, they often neglect local contextual information. Additionally, DLinear [10] and the experiments in Section 4.6 show that transformer-based models are generally ineffective at capturing global temporal correlations in time series. Moreover, using the self-attention mechanism for feature extraction at the core of transformer-based models leads to high computational costs.

The latest trend in TSF research has begun to challenge the efficiency issues of Transformer-based models in prediction tasks. Models like TSMixer [11], LightTS [12] and DLinear [10], which employ simple linear structures, have shown superior performance in TSF compared to the majority of Transformer-based models. However, these models focus on extracting global correlation information in the horizontal time domain, neglecting the understanding

of local contextual information and nonlinear information between different time observations, which is crucial for TSF.

Recently, prediction models based on convolutional modules have shown outstanding performance in TSF [13, 14]. These models typically utilize convolutional layers to extract local nonlinear features from the time series, which to some extent preserve the dynamic information of the time series. Common methods used for extracting local features include standard convolution and dilated convolution (Figure 2 a, b). The receptive field of standard convolution typically covers only a fixed-size region within the input sequence, exhibiting regularity and fixedness [15]. Dilated convolution [14] extends standard convolution by introducing a dilation factor in the convolution kernel, allowing the kernel to span larger receptive fields to capture multi-scale features. However, they possess symmetric receptive fields [16], which limits the ability of convolutional layers to perceive features between asymmetric variables, thus presenting certain limitations in capturing nonlinear characteristics and dynamics of sequences.

To address the aforementioned challenges, we focus on the effective extraction of time series features. Specifically, to capture the temporal domain information of the time series, we consider extracting local features and patterns at different resolutions within the time series. To extract nonlinear features between different time observations in the time series, we improve deformable convolutions [17] (Figure 2 c), which are commonly used in the field of image processing. Notably, directly applying deformable convolutions to time series prediction tasks may lead to issues of data mismatch and ineffective learning of temporal intervals.

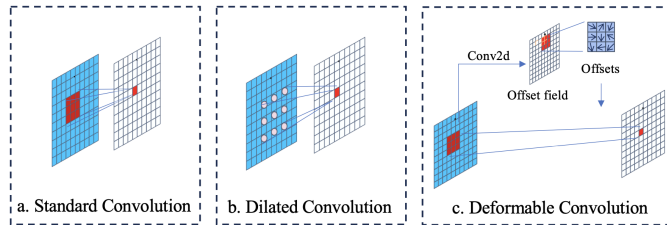


Figure 2: Multiple convolution schemes.

In this work, we propose ACNet, a framework with a multi-resolution mixing architecture capable of extracting local and global information from complex time series data through a temporal feature extraction module (TFE), and then extracting nonlinear features and patterns through a nonlinear feature adaptive extraction module (NFAE). Finally, it predicts future sequences rapidly using a feedforward network and the pseudo-inverse algorithm. With our meticulously designed architecture, ACNet consistently demonstrates leading performance across all experiments, showing superior efficiency in both long-term and short-term forecasting tasks, and covering a wide range of well-established benchmarks. The main contributions are as follows:

- We propose ACNet, an adaptive convolutional network for long-term and short-term time series forecasting. Experimental studies on fifteen real-world datasets demonstrate that the proposed model outperforms the state-of-the-art ConvTimeNet/Crossformer models by 63.4% and 12.5% in long-term and short-term time series forecasting, respectively.
- To extract temporal patterns from the time series, we use convolutions with different dilation factors to extract local contextual features at different resolutions of the input sequence. By overlaying feature information of different resolutions at the same position, we ensure the positional invariance of the input sequence. By employing adaptive average pooling operations, the model gains the capability to extract global feature information from the time series, aiding in capturing overall patterns and trends within the sequence.
- To extract nonlinear features and patterns from complex time series data, we designed a nonlinear feature adaptive extraction module. This module employs improved deformable convolutions, which adaptively adjust the shape of the convolutional kernel based on the input time series features, enhancing the model's ability to capture nonlinear characteristics and complex patterns between variables in complex time series, thereby improving the accuracy of time series predictions. Additionally, deformable convolutions allow the network to adaptively modify the convolutional kernel shape according to specific data features, accurately capturing the nonlinear features and patterns between different time observations in the time series, and significantly reducing the extraction of redundant features.
- To improve the computational efficiency of the model, we combine a single hidden layer feedforward neural network (SLFN) with the pseudo-inverse algorithm, which allows for the rapid computation of the optimal solution, significantly reducing training time and computational resource consumption. This approach ensures that our model maintains high prediction accuracy while achieving greater computational efficiency.

The remaining structure of this paper is as follows: In Section 2, we introduce existing methods for time series forecasting and the application of deformable convolutions. Section 3 provides a detailed description of the ACNet architecture. Section 4 presents experimental results validating the effectiveness and efficiency of ACNet. Finally, in Section 5, we provide a comprehensive summary of the predictive research on the ACNet model.

2. Related work

In this section, we delineate the related work into two distinct modules. Section 2.1 provides an overview of prominent contributions within the field of time series forecasting, while Section 2.2 delves into the applications of deformable convolutions in other fields.

2.1. Time modeling in time series forecasting

Deep learning models have achieved significant success in time series forecasting, and they can be primarily categorized into three paradigms: CNN-based models, Transformer-based models, and MLP-based models.

CNN-based models utilize convolutional kernels along the temporal dimension to effectively capture local temporal patterns. For instance, TCN [18] uses one-dimensional dilated convolutions to expand the receptive field and performs well in both short-term and long-term forecasting. MICN [14] employs multi-scale convolution to extract both global and local contextual relationships. SCINet [19] decomposes time series into multiple resolutions dynamically through SCI-Blocks in a recursive downsampling fashion to extract features at multiple scales. TSLANet [20] learns long-term and short-term relationships in the data through convolutional operations. ConvTimeNet [13] adaptively segments time series into patches and integrates deepwise convolution and pointwise convolution operations to capture global sequence dependencies and cross-variable interactions. However, the aforementioned models are limited to extracting nonlinear features within a fixed receptive field, which hinders their ability to capture nonlinear features among asymmetric variables.

Transformer-based models have garnered widespread attention due to their capability in capturing long-range dependencies through attention mechanisms. For instance, models such as Autoformer [21], ETSformer [22] and FEDformer [23] decompose time series data to extract complex patterns. CrossFormer [9] captures temporal and cross-dimensional dependencies in multivariate time series data by first transforming it into a 2D array and then employing two-stage attention layers. PatchTST [7] employs a channel independence strategy specifically designed to extract correlations of each channel in multivariate data. Pathformer [24] divides the time series into patches of different scales and designs attention mechanisms within and between patches. iTransformer [8] effectively captures multivariate correlations by independently embedding each variable in the time series as variable sub-tokens. However, as the forecasting horizon increases, the attention mechanism's extraction capability may diminish.

Models based on MLPs utilize simple linear models to extract abstract representations of time series, such as LightTS [12], DLinear [10] decomposes time series and utilizes linear network layers for modeling to achieve prediction. TSMixer [11] combines Patch and MLP, enabling it to extract both temporal correlations and inter-channel correlations. FreTS [25] and TimeMixer [26] conduct long-term and short-term forecasting by decoupling historical information of complex patterns in multiscale time series. However, when confronted with high-dimensional data, the expressive capacity of linear networks is constrained, thereby impeding their ability to accurately capture nonlinear features within complex and noisy time series.

2.2. Application of deformable convolutions

Deformable convolutions have found widespread application in the field of image recognition. Recent research has increasingly combined deformable convolutions with attention mechanisms. For instance, in [27, 28], researchers utilized deformable convolutions to capture significant offset information in specific spatial structures, thereby enhancing recognition precision. In [29], due to the complex geometric transformations and feature blurring present in the data, A2-DCNet modules were employed to capture remote spatial context information from a global perspective. Additionally, [30] demonstrated that attention blocks guided by deformable convolutions could acquire semantic information about spatial positions.

Furthermore, research utilizing deformable convolutions in object detection tasks has yielded promising results [31]. In [32], a background-guided deformable convolutional autoencoder network was proposed, effectively separating anomalies from complex backgrounds and enhancing anomaly detection capabilities. Moreover, [33] showed that stacking deformable convolutions and integrating semantic segmentation improved the understanding of contextual relationships. Finally, [34] illustrated that deformable convolutions, akin to spatial attention, could be employed for multiscale feature extraction, while channel attention was used to identify significant features.

To address the aforementioned challenges, we developed a temporal feature extraction module aimed at capturing local contextual information and global patterns within time series data. Additionally, we were inspired by research on the application of deformable convolutions in other domains, prompting us to explore the possibility of introducing them into the field of TSF. By incorporating deformable convolutions, we are able to adaptively capture nonlinear features and patterns within time series data, thereby improving prediction accuracy and performance.

3. Methodology

In this section, we will provide a detailed overview of the ACNet architecture. As depicted in Figure 3, the ACNet model is designed to effectively extract the time-domain pattern characteristics of time series and complex nonlinear information between different observations from multi-dimensional historical data to achieve accurate prediction of time series. The prediction process of the ACNet model mainly consists of the following three stages:

(1) Data processing: For a given multidimensional input dataset, first, standard normalization is applied to eliminate the influence of different resolutions, enhancing the robustness of the model. Subsequently, wavelet denoising is performed on the data to alleviate the interference of noise on model predictions.

(2) Feature acquisition: Fully extracting and utilizing the correlation information and nonlinear features between different time observations in a time series is key to accurately predicting TSF. We employ a temporal feature extraction module and a nonlinear feature adaptive extraction module to extract explicit correlated features hidden in the raw samples, which can significantly enhance prediction accuracy.

(3) Dynamic prediction: As time progresses, the distribution of time series data in real-world scenarios is likely to change, which may lead to decreased predictive accuracy of the model. Therefore, to maintain the model's performance, it is necessary to recalibrate the model parameters when there is a decline in predictive accuracy. This adjustment helps adapt to the features of new samples, ensuring the model's effectiveness in dynamic forecasting tasks.

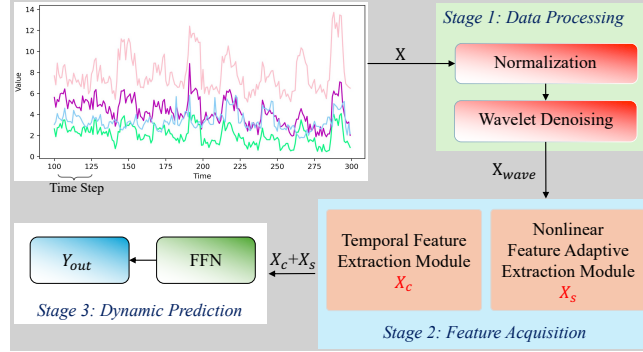


Figure 3: Framework of the ACNet.

3.1. Problem statement

In a rolling prediction setup with a fixed-size window, we consider an input sequence denoted as $X = \{x_1, \dots, x_i, \dots, x_m\} \in \mathbb{R}^{M \times N}$, where $x_i = \{x_i^1, \dots, x_i^j, \dots, x_i^N\}$, and x_i^j represents the value of variable j at the i^{th} time point. The objective of long-term prediction for multivariate time series is to forecast $x_{m+L_y} = \{x_{m+1}, \dots, x_{m+L_y}\}$, where the output length L_y represents an extended future time period.

3.2. Data processing

To enhance the robustness of the model, we normalize the data and convert it into time series windows for training and testing. We employ a normalization method to adjust the time series data to a unified scale, expressed as

$$F(i) = \frac{x_i - \mu}{\sigma}, \quad (1)$$

where μ and σ are the mode-wise mean and variance vectors in the training time series.

The accuracy of the model is constrained by the quality of the dataset, which directly impacts the model's predictive outcomes. To enhance the data quality of training samples and mitigate the influence of noisy data on model training, we employ a compromise method between soft and hard thresholds [35]. This method involves filtering based on calculating the optimal denoising threshold, aiming to achieve effective noise reduction.

We assume the presence of a set of data affected by Gaussian white noise, which can be expressed using the following formula:

$$f_i = \theta_i + \varepsilon e_i, \quad (2)$$

where i is the i^{th} sample, $i = 1, 2, \dots, m$, e_i is white noise, θ_i is the data after denoising, and ε is the level of noise.

The following is a detailed description of the wavelet transform thresholding denoising process:

First, perform an orthogonal wavelet transform and select X historical data samples as the input for discrete wavelet transformation. Then, decompose the data into wavelet coefficients up to the j^{th} layer. The wavelet decomposition coefficients $O_{j,k}$ for each layer can be calculated using the following formula:

$$O_{j,k} = \langle f, \Psi_{j,k} \rangle = \int_{-\infty}^{+\infty} f(x) \Psi_{j,k}(t) dt, \quad (3)$$

where f represents the input sequence.

Next, the decomposed wavelet coefficients are subjected to thresholding, where each coefficient is compared with a predefined threshold to obtain the estimated wavelet coefficients. The soft-hard threshold compromise denoising method is expressed as

$$\widehat{O}_{j,k} = \begin{cases} \text{sgn}(O_{j,k}) (|O_{j,k}| - a\gamma), & |O_{j,k}| \geq \gamma, \\ 0, & |O_{j,k}| < \gamma, \end{cases} \quad (4)$$

where $O_{j,k}$ represents the wavelet coefficients obtained from signal decomposition, $\widehat{O}_{j,k}$ represents the wavelet coefficients obtained using the compromise threshold method, γ represents the threshold value of wavelet denoising, $0 \leq a \leq 1$ can obtain a better denoising effect.

Finally, wavelet reconstruction is performed on the estimated wavelet coefficients by using inverse wavelet transform to obtain denoised samples.

$$\widetilde{X}(t) = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} \widehat{O}_{j,k} \Psi_{j,k}. \quad (5)$$

3.3. Feature acquisition

To comprehensively and effectively extract features from time series, we designed the feature acquisition module, which consists of two components: the temporal feature extraction module and the nonlinear feature adaptive extraction module, as shown in Figure 4 (a). The former is utilized to capture temporal dependency information of local and global patterns within the time series (as shown in the *TemporalFeatureExtractionModule* in

Figure 4 (a)), while the latter is employed to effectively extract nonlinear information between different time observations and correlations among different variables at various resolutions in the time series. (as shown in the *Nonlinear Feature Adaptive Extraction Module* in Figure 4 (a)).

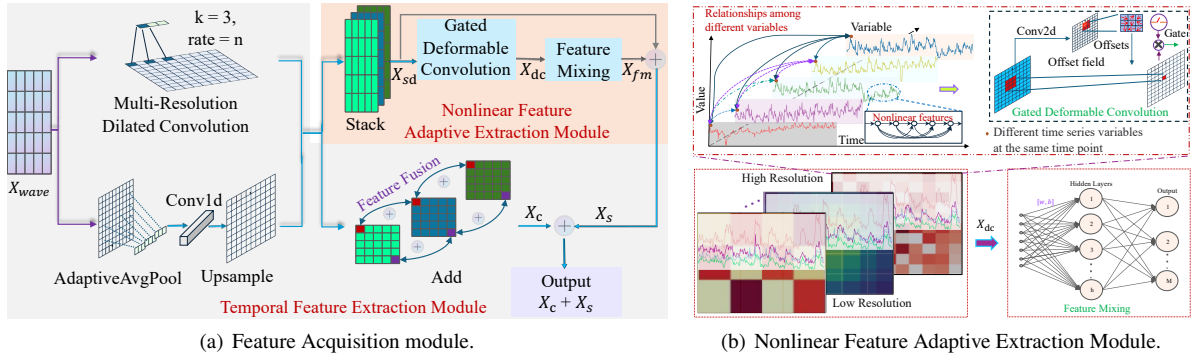


Figure 4: Temporal-Nonlinear feature extraction module.

Temporal Feature Extraction Module Inspired by the concept of SPP [36], the temporal feature extraction module employs one-dimensional (1D) convolutions with varying dilation factors to capture features and patterns at different resolutions within the input multivariate time series data. This approach equips the model with the capability to discern local feature variations: with small dilation factors, the model convolves within a small receptive field, facilitating more precise identification of local patterns and structures, thus aiding in enhancing the network's comprehension of complex patterns. Conversely, with large dilation factors, the model employs larger convolutional kernels to capture features within a broader receptive field, thereby facilitating the identification of local contextual relationships within the input sequence.

Additionally, by aggregating feature and pattern information extracted at different resolutions at the same location, it ensures positional invariance of the data during the processing. To encapsulate global trends and patterns, our model incorporates adaptive average pooling operations. This technique is instrumental in distilling the overarching trends and patterns from the time series data, thus enriching the model's comprehensive grasp of the temporal dynamics presented.

When data flows through this module, it is mapped into two tasks: (1) Local feature extraction. By selecting different dilation factors for multi-resolution dilated convolution, the receptive field of the convolutional kernel can be expanded without losing resolution. This module utilizes convolutions of different sizes to learn patterns and local contextual information at different resolutions. (2) Global feature extraction. Introducing global adaptive average pooling effectively captures global information from the entire feature map, aiding the model in better understanding the global patterns and structural information of the entire input sequence. In summary, by overlaying features of different resolutions at the same location, this not only enhances the model's translational invariance but also improves

the model's generalization capability through the integration of complementary information. Furthermore, through the designed weight sharing mechanism, the same set of weights is reused across inputs of different resolutions. This not only reduces the number of parameters in the model but also helps to enhance the model's efficiency.

The calculation formula for specific operational steps is as follows:

(1) Given a time series X_{wave} with a length of L , a convolution kernel w with a length of k , and a dilation factor d (by default, $d = 1$), the formula for calculating the 1D dilated convolution is as follows:

$$Conv1d = \sum_{i=1}^k w_i x(t + d \times i), \quad (6)$$

where $Conv1d$ represents the output value at time t , $x(t)$ represents the t -th element in the input sequence, w_i represents the i th element of the convolution kernel w . The calculation formula indicates that at each time t , the output value $Conv1d$ is obtained by weighting and summing the local area of x .

(2) Temporal feature extraction.

$$\begin{aligned} \chi_{dilated} &= Relu(LN(Conv1d(\chi_i^l, d = n))), n \in R, \\ \chi_{avg_pool} &= Upsample(Conv1d(AvgPool(\chi_i^l))). \end{aligned} \quad (7)$$

In the equations, $\chi_{dilated}$ is the result of processing the original data χ_i^l through convolution, linear normalization (LN) and ReLU activation function processing. To balance performance and efficiency, we set the dilated rate $n = \{1, 2, 5\}$ as suggested in [37]. χ_{avg_pool} is the result of adaptive average pooling after up sampling.

Through the aforementioned operations, we capture the local features and pattern information of the time series at different resolutions in $\chi_{dilated}$ and the global information of the time series in χ_{avg_pool} . Then, at the corresponding positions, features from different resolutions at the same time point are fused to obtain the final representation X_c of cross-scale temporal feature information. This design allows the model to obtain contextual information from different temporal scales, providing more perspectives for inference and prediction while ensuring robustness and generalization.

$$X_c = Add(\chi_{dilated}, \chi_{avg_pool}). \quad (8)$$

Nonlinear Feature Adaptive Extraction Module Figure 1 illustrates that dependencies between different time observations in time series data may exhibit nonlinear and non-uniform distributions. To extract nonlinear features from complex time series, we designed a gated deformable convolution (GDC) module. Specifically, we added a gating mechanism to the deformable convolution, which further enhances the model's flexibility and capability, helping it to better capture nonlinear features in complex time series. By adjusting the positions of the convolutional kernels, deformable convolution enhances the model's ability to capture local nonlinear features, better handling irregular

features and patterns, thereby improving its perception of local nonlinear patterns, as shown in Figure 4 (b). This capability is particularly important for time series data with significant nonlinear and dynamic feature changes.

Specifically, we stack the extracted multi-resolution feature maps and the output maps from adaptive average pooling into a tensor X_{sd} , which serves as the input to the nonlinear feature adaptive extraction module. We employ GDC module to learn the nonlinear information between different observations and position offset information in feature graphs with different resolutions, focusing attention on regions with richer information through twist sampling networks.

This operation enables the model to more accurately capture and understand the nonlinear features and relationships between different variables that vary between short-term and long-term in time series, while removing redundant feature information. The mathematical expression of the nonlinear feature extraction module is as follows:

$$DefConv(p_0) = \sum_{p_n \in R} \mathcal{W}(p_n) \cdot x(p_0 + p_n + \Delta p_n), \quad (9)$$

where $DefConv(p_0)$ is the value of the output feature map at location p_0 . $x(\cdot)$ represents the input feature map. $\mathcal{W}(p_n)$ denotes the weights of the convolutional kernel. Δp_n is the learned offset for each p_n , differing for each point, allowing the convolutional kernel to adaptively adjust its sampling positions on the input feature map. For more detailed theoretical explanations regarding deformable convolutions, please refer to reference [17]. The nonlinear feature adaptive extraction is described by the following equations:

$$\begin{aligned} X_{sd} &= Stack(\chi_{dilated}, \chi_{avg_pool}), \\ Gate &= Sigmoid(Conv2D(X_{sd})), \\ X_{dc} &= Gate * DefConv(X_{sd}), \end{aligned} \quad (10)$$

where $Conv2D$ denotes a two-dimensional convolution operation, while $sigmoid$ represents the activation function.

Additionally, we implemented a feature mixing layer to integrate the lower-level fine-grained time series information with the higher-level coarse-grained time series information, aiming to merge the correlation information between features at different resolutions. Finally, the use of residual connections allows important feature information to propagate between network layers, preventing information loss. The process is described as follows:

$$\begin{aligned} X_{fm} &= Linear(X_{dc}), \\ X_s &= X_{sd} + X_{fm}. \end{aligned} \quad (11)$$

In the end, the correlations between different dimensions in the time series are captured in X_s .

3.4. Dynamic prediction

Model prediction. In the downstream prediction tasks of ACNet, we employ a simple feedforward neural network (FFN) architecture. To address potential underfitting issues in regression tasks and improve the computational efficiency of the model, we utilize the Moore-Penrose pseudo-inverse matrix [38] to compute the parameters of the model network quickly. The mathematical expression for the model prediction task is as follows:

$$H(G) = [h_1(x_1), h_2(x_2), \dots, h_L(x_L)], \quad (12)$$

where $G = X_c + X_s$, L is the number of hidden layer nodes, $h_i(x_i) = \text{sigmoid}(w_i x_i + b_i)$ represents the output of the i -th hidden layer node, and $H(G)$ is the output matrix of the hidden layer. The weight matrix β of the output layer is calculated by the least squares method:

$$\beta = (H(G)^T H(G))^{-1} H(G)^T Y, \quad (13)$$

where Y is the target matrix. Ultimately, the output y_i predicted by the model is

$$y_i = \sum_{j=1}^L \beta_j h_j(x_i) = H(G) \beta. \quad (14)$$

Model dynamic updates. As time progresses, the data distribution in real-world applications may change, resulting in a significant deterioration of the model's predictive performance. To address this issue, we extend the model training process to a dynamic update stage. We assume that when the model's predictive performance decreases by 5%, we adopt newly collected samples to retrain the ACNet model. It is noteworthy that during this process, we maintain a fixed training set size, which means that when introducing new data, an equal amount of the earliest historical data is removed. This strategy ensures the speed of model training and reduces the required time.

Algorithm 1 outlines the details of the dynamic prediction process.

4. Experiments

In this section, we will delve into the effectiveness and efficiency of the ACNet network. Through rigorous experiments on long-term and short-term forecasting using twelve real-world datasets and comparing against fifteen SOTA models, our goal is to address the following research questions:

Algorithm 1 : Dynamic prediction.**Input:**

$X = \{x_1, \dots, x_i, \dots, x_m\}$: X denotes the training dataset, where x_i represents the N variable factors at time i .

- 1: **Phase I.** Training the proposed ACNet model.
- 2: **Randomly generated:** The weight ω and the bias vector b between the input layer and the hidden layer in FFN.
- 3: Data processing X_{wave} is obtained based on formulas (1) to (5).
- 4: Extract temporal correlation X_c based on formulas (7) to (8)
- 5: Extract inter-variable correlation X_s based on formulas (10) to (11)
- 6: Input $X_c + X_s$ into the FFN and calculate β .
- 7: **Phase II.** Dynamically forecasting the values of a time series.
- 8: Input the obtained new samples x_j and training data set X into the trained ACNet model. If the MSE value exceeds the threshold, update β .

Output: Predicting the trend of changes over a future time period.

- RQ1 (Effectiveness): Can ACNet outshine the present SOTA baseline models when applied to real datasets (Sections 4.4 - 4.8)?
- RQ2 (Efficiency): Does ACNet exhibit superior performance in terms of resource utilization compared to the current baseline models (Sections 4.9)?
- RQ3 (Ablation): To what extent do the distinct components of ACNet influence its overall performance in time series forecasting tasks (Sections 4.10)?

4.1. Datasets

We extensively evaluated the proposed ACNet model on twelve benchmark datasets spanning four real-world fields: energy, finance, medical, and traffic. Table 1 summarizes the key features of these datasets.

- **ETT¹**: The dataset records continuous operation data of power resources over a period of two years, including seven indicators such as oil temperature and load.
- **Electricity²**: The dataset comprises hourly electricity consumption data for 321 customers over two years.
- **Traffic³**: The dataset consists of measurements collected every hour from 862 sensors located on the highways in the San Francisco Bay Area.
- **Exchange⁴**: The dataset records the daily exchange rates of eight different countries over a period of 26 years.
- **ILI⁵**: The dataset collects seven indicators, including the proportion of influenza patients and the total number of patients, from the Centers for Disease Control and Prevention in the United States on a weekly basis.

¹ETT dataset was acquired at <https://github.com/zhouhaoyi/ETDataset>

²[https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams 20112014](https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams+20112014)

³<http://pems.dot.ca.gov>

⁴<https://github.com/laiguokun/multivariate-time-series-data>

⁵<https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html>

Table 1

The key features of the twelve time series datasets.

Tasks	Datasets	Dim	Timesteps	Granularity	Information
Long-term Forecasting	ETTh1/h2	7	17420	1 hour	Electricity
	ETTm1/m2	7	69680	15 min	Electricity
	Electricity	321	26304	1 hour	Electricity
	Traffic	862	52560	1 hour	Traffic
	Exchange	8	7558	1 day	Financial
	ILI	7	966	1 week	Medical
Short-term Forecasting	PEMS03	358	26208	5 min	Traffic
	PEMS04	307	16992	5 min	Traffic
	PEMS07	883	28224	5 min	Traffic
	PEMS08	170	17856	5 min	Traffic

- **PEMS[39]:** This dataset records traffic network data collected every 5 minutes on California highways, including metrics such as traffic flow, speed, and occupancy.

4.2. Baselines

We selected 15 representative SOTA models that have demonstrated outstanding performance in the field of TSF as benchmarks: Patch-based TSMixer [11], PatchTST [7], and PDF [40]; MLP-based TimeMixer [26], FITS [41], and DLinear [10]; Transformer-based architectures iTransformer [8], Crossformer [9], ETSformer [22], FEDformer [23], and Non-Stationary Transformer [42]; and CNN-based models ConvTimeNet [13], TSLANet [20], MICN [14], and TimesNet [43].

4.3. Implementation details

Our approach is trained with an initial learning rate of 1×10^{-3} , the convolution dilated rate d for multi-resolution dilated convolution was set to $\{1, 2, 5\}$. In long-term forecasting, the input sequence length for the ILI dataset is $L = 36$, with prediction sequence lengths of $\{24, 36, 48, 60\}$. For other datasets, the input sequence length is $L = 96$, with prediction sequence lengths of $\{96, 192, 336, 720\}$. In short-term forecasting, the input sequence length for the PEMS datasets is $L = 96$, with a prediction sequence length of 12. To verify the robustness of our results, we conducted long-term forecasting by training the ACNet model with three different random seeds. For each seed, we calculated the mean squared error (MSE) and mean absolute error (MAE). The average results are presented in Table 2. For short-term forecasting, we employed mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean squared error (RMSE) as evaluation metrics. Additionally, we assessed the model's efficiency based on floating point operations (FLOPs), number of parameters, training time, and inference time. Experiments are implemented using PyTorch on a single NVIDIA GeForce RTX 3090 24GB GPU, Intel(R) Core(TM) i7-10700k CPU and 32GB RAM.

4.4. Main results and analysis

Long-term forecasting. In the field of long-term forecasting, the ACNet framework demonstrates excellent performance on almost all datasets. Through analysis of the result data in Table 2, the following conclusions can be drawn:

- The ACNet model significantly improves inference performance on nearly all datasets. As the forecasting horizon increases, the prediction error of ACNet shows a gradually stable upward trend, indicating its ability to maintain high long-term robustness in practical applications.
- Compared to the current SOTA forecasting model ConvTimeNet, ACNet exhibits notable improvements across various datasets. Specifically, ACNet gives 47.9% ($0.319 \rightarrow 0.166$) MSE reduction in ETTm1, 36.9% ($0.211 \rightarrow 0.133$) in ETTm2, 42% ($0.395 \rightarrow 0.229$) in ETTh1, 58% ($0.436 \rightarrow 0.183$) in ETTh2, 32.1% ($0.202 \rightarrow 0.137$) in Electricity, 39.8% ($0.402 \rightarrow 0.242$) in Traffic, 65.7% ($0.347 \rightarrow 0.119$) in exchange and 82.9% ($1.866 \rightarrow 0.319$) in ILI. Overall, ACNet yields a 63.4% averaged MSE reduction among above settings.

Notably, ACNet has achieved great success on the Exchange and ILI datasets, which contain a large number of nonlinear features. This demonstrates the effectiveness of the ACNet model in handling nonlinear features. Notably, ACNet achieved significant success on the Exchange and ILI datasets, which contain a large number of nonlinear features. Compared to the ConvTimeNet model, which uses depthwise separable convolutions to extract time series features, ACNet's extraction of nonlinear features and inter-variable correlations at different resolution scales proves to be more advantageous for TSF.

- ACNet outperforms both the iTransformer and TSMixer models significantly. Compared to iTransformer, ACNet reduces the MSE across all datasets by 66.8% ($0.576 \rightarrow 0.191$). Similarly, compared to TSMixer, ACNet reduces the MSE across all datasets by 49.6% ($0.379 \rightarrow 0.191$). iTransformer and TSMixer both emphasize feature extraction between different variables in time series. While they perform better than models such as PatchTST, which focuses solely on extracting temporal information, they may be affected by strong correlations between different variables, leading to information redundancy and decreased prediction accuracy. ACNet uses deformable convolution to extract the correlation between different variables in the time series and uses a flexible warp network to focus on important features, remove redundant features, and improve the prediction performance of the model.
- The ACNet model achieves significantly better results than the SOTA MLP-based models. ACNet has an average MSE decrease of 67.5% ($0.587 \rightarrow 0.191$) on all datasets in TimeMixer and 49.6% ($0.379 \rightarrow 0.191$) in TSMixer. We analyzed that this may be because MLP-based models need to extract more feature information from complex

Table 2

Long-term multivariate forecasting results with different prediction lengths. The best results are in bold numbers. Avg represents the average value across the four prediction lengths.

Models		ACNet (Ours)	ConvTimeNet (2024)		TimeMixer (2024)		iTransformer (2024)		FiTS (2024)		PDF (2024)		TSMixer (2023)		PatchTST (2023)		MICN (2023)		TimesNet (2023)		Crossformer (2023)		DLinear (2023)		ETSformer (2022)		FEDformer (2022)		Non-Stationary (2022)		
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE		
ETm1	96	0.142	0.263	0.264	0.330	0.320	0.357	0.334	0.368	0.351	0.370	0.320	0.351	0.319	0.358	0.324	0.361	0.316	0.362	0.338	0.375	0.428	0.444	0.345	0.372	0.375	0.398	0.379	0.419	0.386	0.398
	192	0.162	0.282	0.316	0.368	0.361	0.381	0.377	0.391	0.392	0.393	0.375	0.376	0.369	0.384	0.362	0.383	0.363	0.390	0.374	0.387	0.445	0.468	0.380	0.389	0.408	0.410	0.426	0.441	0.459	0.444
	336	0.174	0.295	0.315	0.378	0.390	0.404	0.426	0.420	0.424	0.413	0.411	0.399	0.402	0.406	0.390	0.402	0.408	0.426	0.410	0.411	0.533	0.519	0.413	0.413	0.435	0.428	0.445	0.459	0.495	0.464
	720	0.184	0.304	0.382	0.425	0.454	0.441	0.491	0.459	0.485	0.448	0.464	0.431	0.464	0.442	0.461	0.438	0.481	0.476	0.478	0.450	0.728	0.655	0.474	0.453	0.499	0.462	0.543	0.490	0.585	0.516
	Avg	0.166	0.286	0.319	0.375	0.381	0.395	0.407	0.410	0.413	0.406	0.393	0.389	0.389	0.398	0.384	0.396	0.392	0.414	0.400	0.406	0.534	0.522	0.403	0.407	0.429	0.425	0.448	0.452	0.481	0.456
ETm2	96	0.111	0.219	0.183	0.265	0.175	0.258	0.180	0.264	0.181	0.264	0.181	0.264	0.175	0.258	0.177	0.260	0.179	0.275	0.187	0.267	0.197	0.321	0.193	0.292	0.189	0.280	0.203	0.287	0.192	0.274
	192	0.129	0.236	0.248	0.305	0.237	0.299	0.250	0.309	0.246	0.304	0.242	0.302	0.240	0.290	0.248	0.306	0.307	0.376	0.249	0.309	0.326	0.375	0.284	0.362	0.253	0.319	0.269	0.328	0.280	0.339
	336	0.140	0.247	0.311	0.345	0.298	0.340	0.311	0.348	0.306	0.341	0.302	0.341	0.301	0.338	0.304	0.342	0.325	0.388	0.321	0.351	0.372	0.421	0.369	0.427	0.314	0.357	0.325	0.366	0.334	0.361
	720	0.152	0.255	0.101	0.396	0.391	0.396	0.412	0.407	0.407	0.397	0.403	0.396	0.404	0.398	0.403	0.397	0.502	0.490	0.408	0.403	0.410	0.448	0.554	0.522	0.414	0.413	0.421	0.415	0.417	0.413
	Avg	0.133	0.239	0.211	0.328	0.275	0.323	0.288	0.332	0.285	0.327	0.282	0.326	0.280	0.306	0.283	0.326	0.328	0.382	0.291	0.333	0.326	0.391	0.350	0.401	0.293	0.342	0.305	0.349	0.306	0.347
ETT1	96	0.203	0.326	0.338	0.368	0.375	0.400	0.386	0.405	0.381	0.391	0.369	0.387	0.381	0.391	0.394	0.408	0.398	0.427	0.384	0.402	0.429	0.440	0.386	0.400	0.494	0.479	0.376	0.419	0.513	0.491
	192	0.228	0.347	0.377	0.385	0.429	0.421	0.441	0.436	0.434	0.422	0.414	0.419	0.433	0.420	0.446	0.438	0.430	0.453	0.436	0.429	0.494	0.482	0.437	0.432	0.538	0.504	0.420	0.448	0.534	0.504
	336	0.238	0.355	0.400	0.404	0.484	0.458	0.487	0.458	0.474	0.446	0.451	0.438	0.472	0.441	0.485	0.455	0.460	0.460	0.491	0.469	0.706	0.625	0.481	0.459	0.574	0.521	0.459	0.465	0.588	0.535
	720	0.245	0.363	0.465	0.439	0.498	0.482	0.503	0.481	0.464	0.463	0.463	0.475	0.445	0.467	0.471	0.495	0.474	0.491	0.509	0.521	0.500	0.708	0.689	0.519	0.516	0.535	0.505	0.507	0.643	0.616
	Avg	0.229	0.348	0.395	0.399	0.447	0.440	0.454	0.448	0.438	0.431	0.429	0.430	0.443	0.431	0.455	0.444	0.445	0.462	0.458	0.450	0.595	0.559	0.456	0.452	0.542	0.510	0.440	0.460	0.570	0.537
ETT2	96	0.147	0.259	0.385	0.396	0.289	0.341	0.297	0.349	0.290	0.339	0.292	0.341	0.289	0.338	0.302	0.348	0.332	0.377	0.340	0.374	0.632	0.547	0.333	0.387	0.340	0.391	0.358	0.397	0.476	0.458
	192	0.168	0.279	0.438	0.425	0.372	0.392	0.380	0.400	0.375	0.388	0.377	0.392	0.375	0.391	0.388	0.400	0.422	0.441	0.402	0.414	0.876	0.663	0.477	0.476	0.430	0.439	0.429	0.439	0.512	0.493
	336	0.192	0.295	0.451	0.437	0.386	0.414	0.428	0.432	0.414	0.425	0.419	0.429	0.425	0.435	0.426	0.433	0.447	0.474	0.452	0.452	0.924	0.702	0.594	0.541	0.485	0.479	0.486	0.487	0.552	0.551
	720	0.223	0.310	0.470	0.461	0.412	0.434	0.427	0.445	0.419	0.437	0.424	0.441	0.435	0.449	0.431	0.446	0.462	0.467	0.462	0.468	1.390	0.863	0.831	0.657	0.500	0.497	0.463	0.474	0.562	0.560
	Avg	0.183	0.286	0.436	0.430	0.364	0.395	0.383	0.407	0.375	0.397	0.378	0.401	0.381	0.403	0.387	0.407	0.411	0.440	0.414	0.427	0.956	0.694	0.559	0.515	0.439	0.452	0.437	0.449	0.526	0.516
Electricity	96	0.131	0.253	0.159	0.263	0.153	0.247	0.148	0.240	0.160	0.240	0.165	0.248	0.146	0.244	0.195	0.285	0.164	0.269	0.168	0.272	0.254	0.347	0.197	0.282	0.187	0.304	0.193	0.308	0.189	0.273
	192	0.135	0.239	0.169	0.265	0.162	0.256	0.162	0.253	0.268	0.378	0.181	0.266	0.163	0.259	0.199	0.289	0.177	0.285	0.184	0.289	0.261	0.383	0.196	0.285	0.199	0.315	0.201	0.315	0.182	0.286
	336	0.138	0.245	0.201	0.285	0.185	0.277	0.178	0.269	0.355	0.452	0.197	0.282	0.180	0.279	0.215	0.305	0.193	0.304	0.198	0.300	0.273	0.364	0.209	0.301	0.212	0.329	0.214	0.329	0.200	0.304
	720	0.240	0.250	0.242	0.318	0.225	0.310	0.225	0.317	0.416	0.498	0.238	0.315	0.216	0.307	0.256	0.337	0.212	0.321	0.220	0.320	0.303	0.388	0.245	0.333	0.233	0.345	0.246	0.355	0.222	0.321
	Avg	0.137	0.242	0.202	0.284	0.182	0.272	0.178	0.270	0.353	0.452	0.195	0.278	0.176	0.272	0.216	0.304	0.187	0.295	0.193	0.295	0.273	0.363	0.212	0.300	0.208	0.323	0.214	0.327	0.193	0.296
Traffic	96	0.228	0.286	0.376	0.265	0.462	0.285	0.395	0.268	0.898	0.572	0.463	0.297	0.483	0.321	0.544	0.359	0.519	0.309	0.593	0.321	0.558	0.320	0.650	0.396	0.607	0.392	0.587	0.367	0.612	0.338
	192	0.239	0.292	0.392	0.271	0.473	0.296	0.417	0.276	0.763	0.522	0.469	0.298	0.490	0.321	0.540	0.354	0.537	0.315	0.617	0.336	0.572	0.331	0.598	0.370	0.621	0.399	0.604	0.373	0.613	0.340
	336	0.243	0.300	0.405	0.277	0.498	0.296	0.433	0.283	0.894	0.608	0.484	0.305	0.506	0.330	0.551	0.358	0.534	0.313	0.629	0.336	0.587	0.342	0.605	0.373	0.622	0.396	0.621	0.383	0.618	0.328
	720	0.256	0.307	0.438	0.294	0.506	0.313	0.467	0.302	1.019	0.646	0.538	0.341	0.538	0.341	0.586	0.375	0.577	0.325	0.640	0.350	0.662	0.359	0.645	0.394	0.632	0.396	0.626	0.382	0.653	0.355
	Avg	0.242	0.296	0.402	0.277	0.484	0.297	0.428	0.282	0.894	0.587	0.472	0.300	0.504	0.328	0.555	0.362	0.542	0.316	0.620	0.336	0.592	0.338	0.625	0.383	0.621	0.396	0.610	0.376	0.624	0.340
Exchange	96	0.097	0.209	0.083	0.196	0.100	0.222	0.086	0.206	0.089	0.210	0.082	0.199	0.082	0.199	0.082	0.201	0.102	0.235	0.107	0.234	0.329	0.440	0.088	0.218	0.085	0.204	0.148	0.237	0.111	0.237
	192	0.103	0.217	0.170	0.291	0.122	0.326	0.177	0.299	0.182	0.303	0.175	0.296	0.176	0.297	0.187	0.307	0.172	0.316	0.226	0.344	0.544	0.586	0.176	0.315	0.182	0.303	0.271	0.335	0.219	0.335
	336	0.117	0.232	0.316	0.404	0.379	0.442	0.331	0.417	0.327	0.414	0.327	0.413	0.337	0.418	0.345	0.427	0.272	0.407	0.367	0.448	1.017	0.786	0.313	0.427	0.348	0.428	0.460	0.476	0.421	0.476
	720	0.158	0.279	0.817	0.679	0.904	0.715	0.847	0.691	0.882	0.696	0.849	0.694	0.809	0.725	0.887	0.708	0.714	0.658	0.964	0.746	1.239	0.912	0.839	0.695	1.025	0.774	1.195	0.769	1.092	0.769
	Avg	0.119	0.234	0.347	0.393	0.309	0.426	0.360	0.403	0.363	0.406	0.359	0.401	0.376	0.410	0.375	0.411	0.315	0.404	0.416	0.443	0.782	0.681	0.354	0.414	0.410	0.427	0.519	0.454	0.461	0.454
ILI	24	0.290	0.311	1.701	0.823	2.122	0.874	2.014	0.999	3.340	1.299	2.585	1.065	0.432	0.407	1.724	0.843	2.684	1.112	2.317	0.934	3.041	1.186	2.398	1.040	2.527	1.020	3.228	0.945	2.294	0.945
	36	0.305	0.336	1.941	0.88																										

Table 3

Short-term multivariate forecasting results with different metrics. The best results are in bold numbers and the second best are highlighted with an underline.

Models		ACNet (Ours)	ConvTimeNet (2024)	TimeMixer (2024)	iTransformer (2024)	FITS (2024)	PDF (2024)	TSLANet (2024)	TSMixer (2023)	PatchTST (2023)	MICN (2023)	TimesNet (2023)	Crossformer (2023)
PEMS03	MAE	0.145	0.245	0.187	0.178	0.348	0.195	0.185	0.198	0.212	0.462	0.190	<u>0.166</u>
	MAPE	1.295	1.898	1.466	1.364	2.656	2.794	1.459	1.567	1.636	2.948	1.455	<u>1.321</u>
	RMSE	0.208	0.361	0.277	0.266	0.468	0.293	0.278	0.277	0.310	0.606	0.290	<u>0.253</u>
PEMS04	MAE	0.183	0.307	0.247	0.235	0.448	0.260	0.272	0.238	0.269	0.935	0.269	<u>0.206</u>
	MAPE	1.550	2.635	2.039	1.931	3.658	2.131	2.179	1.985	2.177	7.799	2.208	<u>1.682</u>
	RMSE	0.311	0.566	0.487	0.481	0.623	0.509	0.526	0.434	0.516	1.298	0.534	<u>0.430</u>
PEMS07	MAE	0.130	0.257	0.187	0.181	0.382	0.194	0.197	0.186	0.211	0.622	0.199	<u>0.150</u>
	MAPE	1.349	2.561	1.907	1.849	3.509	2.004	1.948	1.970	2.058	4.773	1.904	<u>1.667</u>
	RMSE	0.191	0.368	0.279	0.274	0.501	0.290	0.297	0.273	0.303	0.800	0.302	<u>0.232</u>
PEMS08	MAE	0.124	0.313	0.259	0.255	0.473	0.268	0.256	0.259	0.285	0.797	0.302	<u>0.238</u>
	MAPE	1.594	2.115	1.649	1.705	3.235	1.704	1.709	1.685	1.800	5.239	2.077	1.497
	RMSE	0.204	0.582	0.523	0.516	0.723	0.531	0.517	0.481	0.540	1.189	0.664	<u>0.497</u>

mechanism to capture global information from the input sequence. Nonlinear relationships, however, often require local contextual information for better understanding and accurate feature extraction.

- The MICN model, which is based on multi-scale dilated convolution, shows the poorest performance across all datasets. This outcome suggests that traditional convolutions with fixed-size receptive fields are inadequate for real-world time series datasets containing numerous nonlinear features. The fixed nature of these receptive fields restricts their flexibility in handling nonlinear features of varying scales, leading to information redundancy and diminished predictive performance.

4.5. Visualization of forecasting

To evaluate the predictive performance of ACNet compared to SOTA TSF models (such as ConvTimeNet, TimeMixer, iTransformer, TSMixer and PatchTST) in real-world scenarios, we qualitatively compared the prediction results of the last dimension on the test set of the ETTh1 dataset, as shown in Figure 5. The input length was set to 96, and the prediction length was set to 192 to assess the fitting between the predicted sequences and the actual sequences.

From Figure 5, it can be observed that ACNet is capable of adapting to the dynamic changes in time series data, capturing both the periodic and trend information. However, while other models effectively capture the periodic information of the time series, they neglect to acquire trend information. For instance, around the 280th time point, the sequence suddenly oscillates downward. ACNet successfully captured the trend information of such non-stationary oscillatory changes, while other models retained the trend information from the previous time period, indicating that their predictive performance did not benefit from the module that extracts long-term feature information. In summary, ACNet can not only capture local contextual information but also effectively capture the global trend information and nonlinear information of time series, thereby achieving better fitting results.

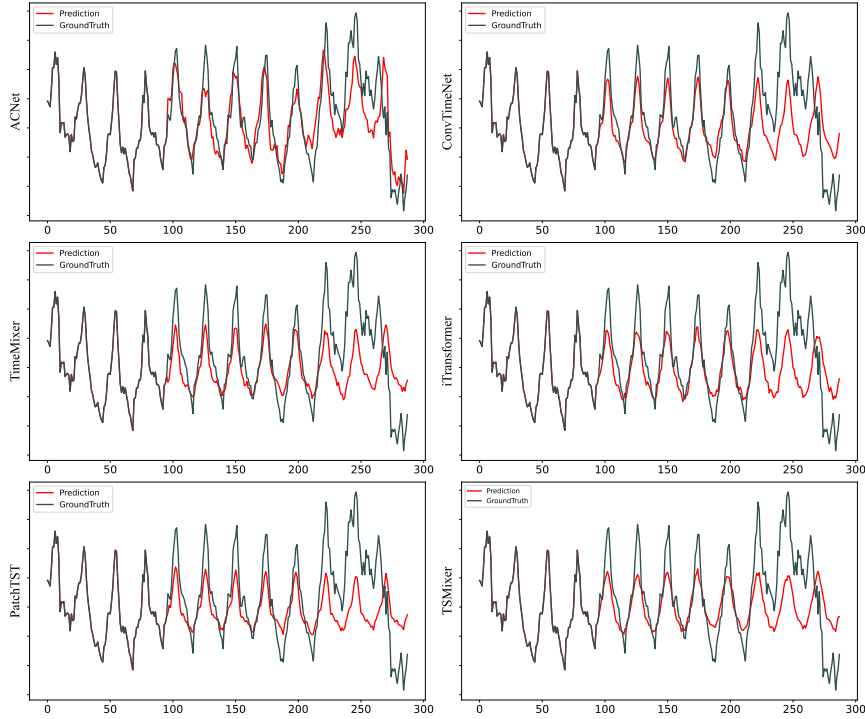


Figure 5: Visualize the ETTh1 dataset, where the first 96 data points represent the input length.

4.6. Long-term information utilization

To validate ACNet's ability to capture long-term correlations, we compared the performance of different models on the ETTh1 and ETTm1 datasets, with output lengths set to {96, 720} and input sequence windows set to {24, 48, 96, 192, 336}. In theory, using longer input sequence windows can increase the receptive field and potentially improve prediction performance. However, as evidenced by the experimental results in Figure 6, the performance of Transformer-based models, TimeMixer, and MICN models does not continuously improve with increasing model input length, indicating that these models do not benefit from longer input sequence windows, i.e., they lack in capturing long-term correlations. In contrast, ACNet, ConvTimeNet, and TSMixer show continuous improvement in model performance (decreasing MSE values) with increasing input sequence length, demonstrating that these models can exploit more features from longer input sequence windows and effectively capture the long-term correlations in the input sequence.

4.7. T-tests

To further validate the effectiveness of the ACNet model compared to other baseline models, we conducted T-tests to assess the significant differences between the prediction results of ACNet and the baseline models. As shown in Figure 7, we can draw the following conclusions: (1) The T-statistic for ACNet relative to all baseline models

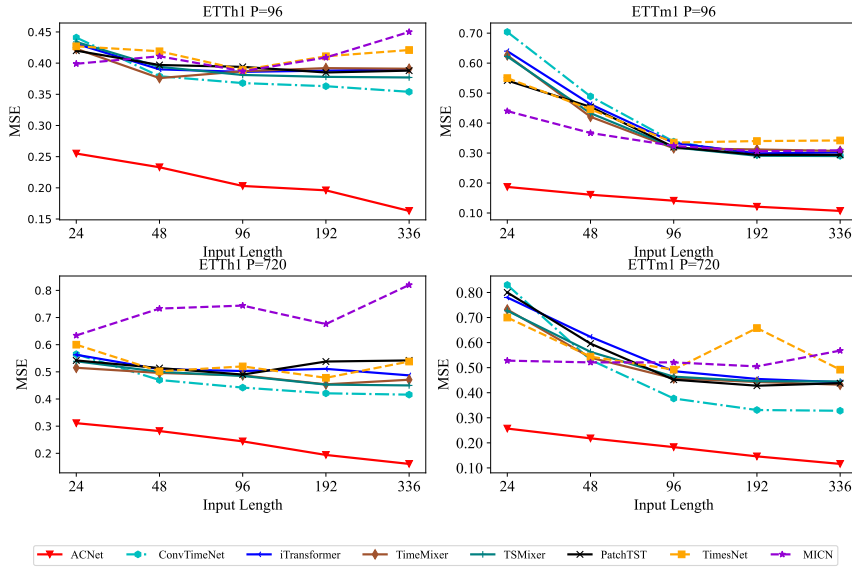


Figure 6: The predictive performance (MSE) of different input sequence windows.

significantly exceeds $T' = 1.782$ (by looking up the critical value in the T-distribution table based on the P-value, we obtain the T-value), indicating that the average MSE of ACNet is significantly lower than that of the baseline models. (2) The calculated P-values are all less than the threshold value of $P = 0.05$, indicating statistically significant differences in prediction results among different models. Based on the above analysis, we can conclude that the observed performance differences between ACNet and the baseline models are indeed the result of true differences, rather than random factors.

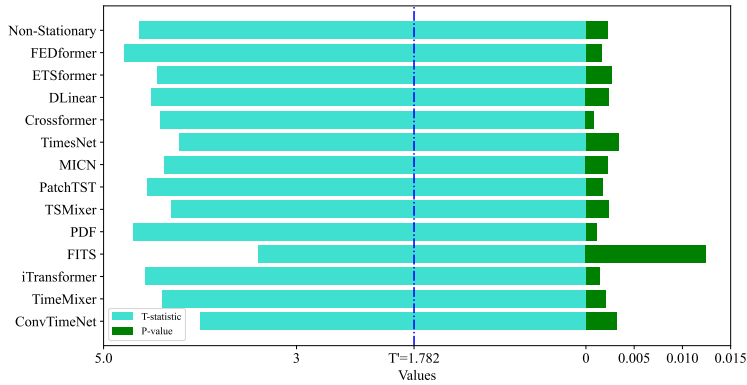


Figure 7: The result of the statistical significance test between ACNet and fourteen baseline models.

4.8. Validation of capturing nonlinear features

To further verify the effectiveness of our proposed nonlinear feature adaptive extraction module in extracting features from real-world time series data, we selected the Exchange dataset for validation. We processed the data using both standard convolutional kernels and our proposed gated deformable convolution (GDC). We used a batch of the dataset and extracted feature maps from the intermediate layers to observe the activation of input data after passing through the convolutional layers. This helps in understanding how different convolutional modules capture nonlinear relationships. We visualized the feature maps from the first eight layers, as shown in Figure 8:

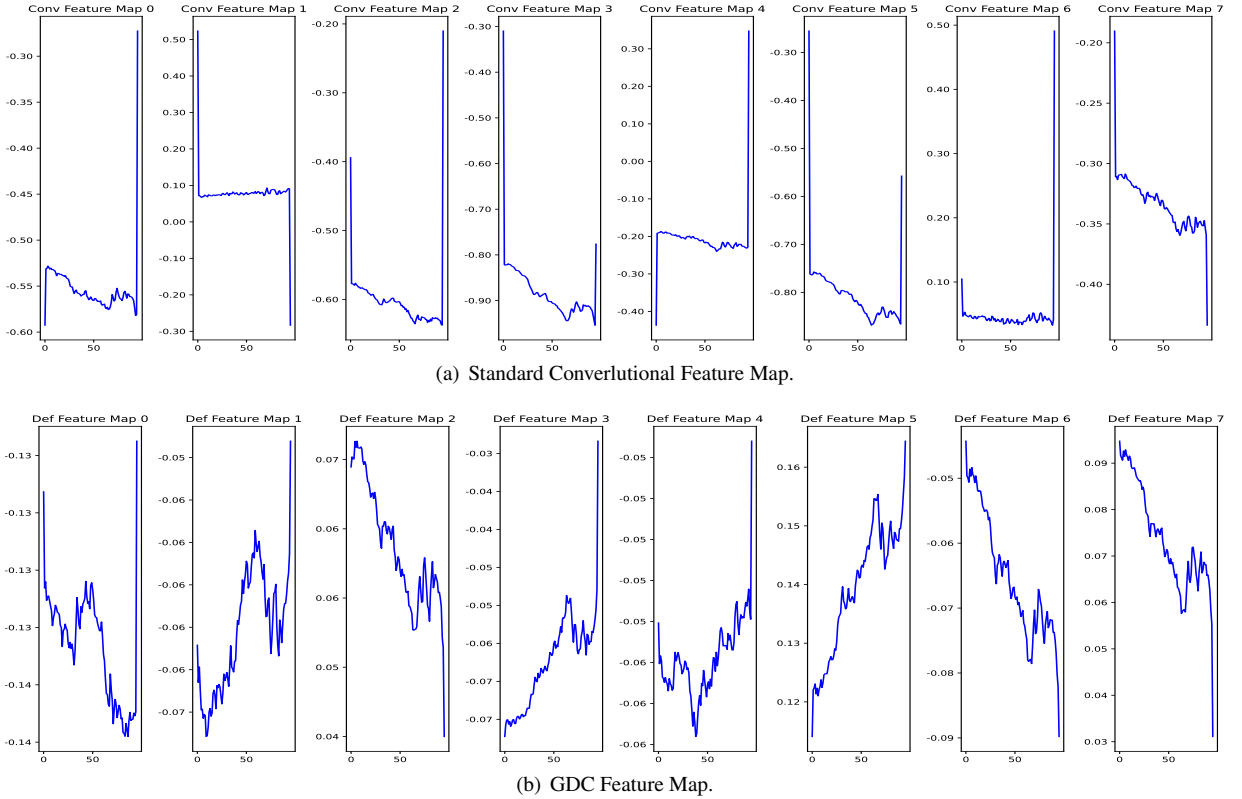


Figure 8: Visualization of feature maps from different convolutions.

From Figure 8, we can draw the following conclusions:

(1) Amplitude of Variations in Feature Maps. The feature maps from standard convolution exhibit smaller amplitudes of change with fewer fluctuations, indicating a potential inadequacy in handling sudden and nonlinear changes in the input data. The feature maps from GDC exhibit larger amplitudes of change with more noticeable fluctuations, indicating a greater flexibility in capturing sudden and nonlinear changes in the input data.

(2) Diversity in Feature Maps. The feature maps from standard convolution show more consistent patterns with relatively smooth changes. The feature maps from GDC show more diverse patterns and variations, indicating a stronger adaptability in handling nonlinear features.

(3) Response to Sudden Changes. The feature maps from standard convolution exhibit smooth transitions at points of sudden changes in the data, lacking sensitivity to such changes. The feature maps from GDC exhibit noticeable fluctuations and variations at points of sudden changes in the data, better capturing these sudden features.

Specifically, from the standard convolutional feature maps (as shown in Figure 8(a)), we can see that most feature maps show relatively smooth and stable changes with almost no significant fluctuations or sudden changes. In feature map 5, some fluctuations can be observed, but the overall changes remain relatively stable. From the GDC feature maps (as shown in Figure 8(b)), we can see that feature maps 1, 3, 4, and 5 exhibit significant fluctuations and sudden changes, indicating that GDC can capture dynamic sudden features in the input data. Feature maps 0 and 2 exhibit more details and fluctuations, suggesting that GDC performs well in handling non-stationary features.

In summary, GDC performs better in handling nonlinear features because it can dynamically adjust the positions of the convolutional kernels, making it more adaptable to variations in the input data.

4.9. Efficiency analysis

To evaluate the efficiency of ACNet, we selected the ETTh1 dataset and compared it with six SOTA models. We conducted efficiency analysis of the models from four aspects: computational complexity (Flops), model size (Parameters), training time, and inference time. To facilitate a clearer comparison and analysis of model efficiency, we integrated these four metrics and computed their average values as the model efficiency scores, as shown in Table 4. Table 4 distinctly demonstrates the efficiency advantages of ACNet, which are particularly crucial for practical applications of TSF. This advantage mainly stems from the model's parameter sharing mechanism, deformable convolutions, and the ability to rapidly update network weights through pseudo-inverse algorithms. Additionally, we observed a decreasing trend in the training time of iTransformer with increasing input length. This is because iTransformer employs an early termination strategy during training, where training halts when the loss value decreases continuously for a certain number of iterations ($n=3$ in iTransformer). According to our experiments, as the input length increases, the model is more likely to meet this condition during training.

4.10. Ablation study

To assess the effectiveness of each component within ACNet, we conducted the following ablations: (1) w/o GDC: remove the gated deformable convolution module used for extracting nonlinear features of time series; (2) w/o Temporal: elimination of modules for extracting temporal correlation information; (3) w/o ALL: complete removal of all feature extraction modules. We conducted ablation studies on the ETTh2 and Electricity datasets, with input lengths set to 96 and prediction lengths set to {96, 192, 336, 720}.

The results in Table 5 indicate that removing any module from the original model led to a decrease in performance. Specifically: (1) Removing the GDC module (w/o GDC) will cause the model to lose its ability to extract nonlinear

Table 4

Results of the efficiency analysis.

Metrics		Flops (G)	Parameters (M)	Training Time (s)	Inference Time (s)	Ranking	
Models						Four Metrics	Avg Ranking
ACNet	192	0.064	0.949	20.308	0.091	(1,4,3,1)	2.25
	384	0.128	1.898	35.999	0.124		
	768	0.254	3.700	49.618	0.217		
	1536	0.510	7.589	73.758	0.436		
ConvTimeNet	192	4.745	0.265	14.978	7.977	(4,1,2,7)	3.50
	384	9.491	0.335	24.781	8.975		
	768	18.981	0.476	38.070	10.97		
	1536	37.962	0.757	48.436	13.962		
iTransformer	192	0.305	0.866	1056.093	2.493	(2,2,7,3)	3.50
	384	0.322	0.915	1036.082	2.658		
	768	0.356	1.014	422.610	2.986		
	1536	0.426	1.210	407.591	3.155		
TimeMixer	192	4.329	0.245	14.209	6.979	(6,5,1,6)	4.50
	384	14.895	0.902	17.310	7.978		
	768	54.757	3.464	27.918	8.576		
	1536	209.391	13.571	54.004	9.474		
TSMixer	192	0.027	0.100	38.392	1.723	(3,3,4,2)	3.00
	384	0.087	0.339	41.811	1.931		
	768	0.306	1.239	56.190	1.976		
	1536	1.139	4.727	69.689	2.200		
PatchTST	192	17.253	4.140	51.121	3.649	(5,6,5,4)	5.00
	384	34.505	5.265	80.290	3.936		
	768	69.010	7.515	87.804	4.057		
	1536	138.021	12.015	101.786	3.955		
MICN	192	108.741	27.542	116.441	2.926	(7,7,6,5)	6.25
	384	187.220	36.559	143.929	3.174		
	768	364.311	48.594	174.319	3.823		
	1536	799.024	76.665	273.936	5.757		

information from the time series. By solely relying on the the time-domain information extracted by the model, the prediction accuracy of the model is reduced. (2) Removing the temporal feature extraction module (w/o temporal) will prevent the model from extracting local contextual information and global patterns from the time series, leading to a decline in predictive performance. (3) Completely removing the feature extraction module (w/o ALL) is equivalent to using only the FFN layer for prediction. Experimental results indicate that this approach performs poorly, once again demonstrating that FFN lacks the ability to effectively extract complex features from time series data.

5. Conclusion

This paper introduces a feature-driven time series prediction network, ACNet. From the perspective of effective feature extraction and utilization, we use convolutional kernels with different dilation factors to capture latent local patterns and features at different resolutions in complex time series. Additionally, adaptive average pooling is employed to learn the global temporal patterns of the time series. By utilizing improved gated mechanism-based deformable

Table 5

Results of the ablation study.

Models		ACNet		w/o GDC		w/o Temporal		w/o All	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh2	96	0.147	0.259	0.156	0.267	0.171	0.276	0.185	0.287
	192	0.168	0.279	0.180	0.287	0.203	0.300	0.226	0.315
	336	0.192	0.295	0.205	0.304	0.236	0.319	0.269	0.337
	720	0.223	0.310	0.237	0.320	0.277	0.340	0.325	0.363
	Avg	0.183	0.286	0.195	0.295	0.222	0.309	0.251	0.326
Electricity	96	0.131	0.233	0.136	0.238	0.141	0.243	0.144	0.246
	192	0.135	0.239	0.144	0.249	0.143	0.247	0.146	0.250
	336	0.138	0.245	0.153	0.255	0.148	0.253	0.151	0.257
	720	0.142	0.250	0.155	0.257	0.150	0.255	0.153	0.258
	Avg	0.137	0.242	0.147	0.250	0.146	0.250	0.149	0.253

convolutions, the network can effectively capture nonlinear features and pattern information in the time series. Considering the inherent variability in the data distribution of time series, the model allows for dynamic updates based on changes in the data. Through extensive experimentation on twelve benchmark datasets, our findings demonstrate the superior performance of ACNet compared to previous SOTA models.

Acknowledgments

This work is partially supported by National Hi-Tech Project, China with grant No. WDZC20215250117.

References

- [1] Y. Lin, Progressive neural network for multi-horizon time series forecasting, *Inf. Sci.* 661 (2024) 120112.
- [2] X. Feng, D. Fan, S. Jiang, J. Zhang, B. Guo, X. Ding, D. Hu, Y. Jiang, A causal representation learning based model for time series prediction under external interference, *Inf. Sci.* 663 (2024) 120270.
- [3] N. Bacanin, L. Jovanovic, M. Zivkovic, V. Kandasamy, M. Antonijevic, M. Deveci, I. Strumberger, Multivariate energy forecasting via metaheuristic tuned long-short term memory and gated recurrent unit neural networks, *Inf. Sci.* 642 (2023) 119122.
- [4] Y. Qin, H. Luo, F. Zhao, Y. Fang, X. Tao, C. Wang, Spatio-temporal hierarchical mlp network for traffic forecasting, *Inf. Sci.* 632 (2023) 543–554.
- [5] S. Mo, H. Wang, B. Li, S. Fan, Y. Wu, X. Liu, Timesql: Improving multivariate time series forecasting with multi-scale patching and smooth quadratic loss, *Inf. Sci.* 671 (2024) 120652.
- [6] S. Huang, Y. Liu, H. Cui, F. Zhang, J. Li, X. Zhang, M. Zhang, C. Zhang, Meaformer: An all-mlp transformer with temporal external attention for long-term time series forecasting, *Inf. Sci.* 669 (2024) 120605.
- [7] Y. Nie, N. H. Nguyen, P. Sinthong, J. Kalagnanam, A time series is worth 64 words: Long-term forecasting with transformers, in: *The Eleventh International Conference on Learning Representations*, 2023.
- [8] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, M. Long, itransformer: Inverted transformers are effective for time series forecasting, *arXiv preprint arXiv:2310.06625* (2023).
- [9] M. Hassanin, A. Khamiss, M. Bennamoun, F. Boussaid, I. Radwan, Crossformer: Cross spatio-temporal transformer for 3d human pose estimation, *arXiv preprint arXiv:2203.13387* (2022).

- [10] A. Zeng, M. Chen, L. Zhang, Q. Xu, Are transformers effective for time series forecasting?, in: Proceedings of the AAAI conference on artificial intelligence, volume 37, 2023, pp. 11121–11128.
- [11] V. Ekambaram, A. Jati, N. Nguyen, P. Sinthong, J. Kalagnanam, Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '23, 2023, pp. 459–469.
- [12] T. Zhang, J. Bian, Y. Zhang, X. Yi, J. Li, W. Cao, S. Zheng, Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures, 2022.
- [13] M. Cheng, J. Yang, T. Pan, Q. Liu, Z. Li, Convtimenet: A deep hierarchical fully convolutional model for multivariate time series analysis, arXiv preprint arXiv:2403.01493 (2024).
- [14] H. Wang, J. Peng, F. Huang, J. Wang, J. Chen, Y. Xiao, Micn: Multi-scale local and global context modeling for long-term series forecasting, in: The Eleventh International Conference on Learning Representations, 2022.
- [15] J. Yang, A. Li, J. Qian, J. Qin, L. Wang, A hyperspectral image classification method based on pyramid feature extraction with deformable-dilated convolution, IEEE Geosci. and Remote Sens. Lett. 21 (2024).
- [16] W. Zeng, C. Lin, K. Liu, J. Lin, A. K. H. Tung, Modeling spatial nonstationarity via deformable convolutions for deep traffic flow prediction, IEEE Trans. on Knowl. and Data Eng. 35 (2023) 2796–2808.
- [17] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017.
- [18] J.-Y. Franceschi, A. Dieuleveut, M. Jaggi, Unsupervised scalable representation learning for multivariate time series, Le Centre pour la Communication Scientifique Directe - HAL - Diderot (2019).
- [19] M. Liu, A. Zeng, M. Chen, Z. Xu, Q. Lai, L. Ma, Q. Xu, Scinet: Time series modeling and forecasting with sample convolution and interaction, volume 35, 2022.
- [20] E. Eldele, M. Ragab, Z. Chen, M. Wu, X. Li, Tslanet: Rethinking transformers for time series representation learning, 2024. arXiv:2404.08472.
- [21] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, volume 27, 2021, pp. 22419–22430.
- [22] G. Woo, C. Liu, D. Sahoo, A. Kumar, S. Hoi, Etsformer: Exponential smoothing transformers for time-series forecasting, arXiv preprint arXiv:2202.01381 (2022).
- [23] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin, Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting, in: International conference on machine learning, 2022, pp. 27268–27286.
- [24] P. Chen, Y. Zhang, Y. Cheng, Y. Shu, Y. Wang, Q. Wen, B. Yang, C. Guo, Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting, 2024.
- [25] K. Yi, Q. Zhang, W. Fan, S. Wang, P. Wang, H. He, D. Lian, N. An, L. Cao, Z. Niu, Frequency-domain mlps are more effective learners in time series forecasting, ArXiv abs/2311.06184 (2023).
- [26] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, J. ZHOU, Timemixer: Decomposable multiscale mixing for time series forecasting, in: The Twelfth International Conference on Learning Representations, 2024.
- [27] L. Dai, G. Zhang, R. Zhang, Radanet: Road augmented deformable attention network for road extraction from complex high-resolution remote-sensing images, IEEE Trans. on Geosci. and Remote Sens. 61 (2023).
- [28] S. Zhuo, J. Zhang, Attention-based deformable convolutional network for chinese various dynasties character recognition, Expert Syst. With Appl. 238 (2024).

- [29] J. Du, W. Fan, C. Gong, J. Liu, F. Zhou, Aggregated-attention deformable convolutional network for few-shot sar jamming recognition, *Pattern Recognit.* 146 (2024).
- [30] J. Luo, P. Huang, P. He, B. Wei, X. Guo, H. Xiao, Y. Sun, S. Tian, M. Zhou, P. Feng, Dca-daffnet: An end-to-end network with deformable fusion attention and deep adaptive feature fusion for laryngeal tumor grading from histopathology images, *IEEE Trans. on Instrum. and Meas.* 72 (2023).
- [31] C. Ma, L. Zhuo, J. Li, Y. Zhang, J. Zhang, Cascade transformer decoder based occluded pedestrian detection with dynamic deformable convolution and gaussian projection channel attention mechanism, *IEEE Trans. on Multimed.* 25 (2023) 1529–1537.
- [32] Z. Wu, M. E. Paoletti, H. Su, X. Tao, L. Han, J. M. Haut, A. Plaza, Background-guided deformable convolutional autoencoder for hyperspectral anomaly detection, *IEEE Trans. on Geosci. and Remote Sens* 61 (2023).
- [33] L. Yu, X. Zhi, J. Hu, S. Zhang, R. Niu, W. Zhang, S. Jiang, Improved deformable convolution method for aircraft object detection in flight based on feature separation in remote sensing images, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sens.* 17 (2024) 8313–8323.
- [34] Y. Shi, C. Wang, S. Xu, M.-D. Yuan, F. Liu, L. Zhang, Deformable convolution-guided multiscale feature learning and fusion for uav object detection, *IEEE Geosci. and Remote Sens. Lett.* 21 (2024).
- [35] C. Wei, Z. Mu, M. W. Bhatt, Railway foreign body vibration signal detection based on wavelet analysis, *J. of Vibroengineering* 24 (2022) 1139–1147.
- [36] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, *IEEE Trans. on Pattern Anal. & Mach. Intell.* 37 (2015) 1904–1916.
- [37] C. Li, Z. Qiu, X. Cao, Z. Chen, H. Gao, Z. Hua, Hybrid dilated convolution with multi-scale residual fusion network for hyperspectral image classification, *Micromachines* 12 (2021) 545.
- [38] J. A. Vásquez-Coronel, M. Mora, K. Vilches, A review of multilayer extreme learning machine neural networks, *Artif. Intell. Rev.* 56 (2023) 13691–13742.
- [39] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, Z. Jia, Freeway performance measurement system: mining loop detector data, *Transportation research record* 1748 (2001) 96–102.
- [40] T. Dai, B. Wu, P. Liu, N. Li, J. Bao, Y. Jiang, S.-T. Xia, Periodicity decoupling framework for long-term series forecasting, in: *The Twelfth International Conference on Learning Representations*, 2023.
- [41] Z. Xu, A. Zeng, Q. Xu, Fits: Modeling time series with 10k parameters, 2024. [arXiv:2307.03756](https://arxiv.org/abs/2307.03756).
- [42] Y. Liu, H. Wu, J. Wang, M. Long, Non-stationary transformers: Exploring the stationarity in time series forecasting, *Adv. in Neural Inf. Process. Syst.* 35 (2022) 9881–9893.
- [43] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, M. Long, Timesnet: Temporal 2d-variation modeling for general time series analysis, in: *The eleventh international conference on learning representations*, 2022.