International Journal of Basic and Applied Sciences, 14 (4) (2025) 98-104

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/tme8bx54
Research paper

A Novel ARM-DC AutoConNet for Accurate Long-Term Time-Series Forecasting

Yangyang Li, Jian Qu *

Faculty of Engineering and Technology, Panyapiwat Institute of Management, Nonthaburi, Thailand
**Corresponding author E-mail: Jianqu@pim.ac.th*

Received: July 24, 2025, Accepted: July 30, 2025, Published: August 5, 2025

Abstract

Time series forecasting is essential in many fields, such as financial analysis, climate forecasting, and energy demand. This study proposes an improved dilated convolutional(DC) AutoConNet architecture to solve the problem of accurate long-term forecasting of complex time series data. The model significantly improves its robustness and generalization ability by integrating multi-scale dilated convolution, layer normalization, and a new adaptive rescaling mechanism(ARM). The main improvement is that while maintaining the efficiency of the original AutoConNet, it effectively solves the overfitting problem and the defect of insufficient capture of temporal dependencies. We evaluated the model performance on 16 standard datasets (including finance, climate, health, etc.), such as M4, M5, ETTh1, ETTh2, ETTm1, and ETTm2. The ARM-DC AutoConNet achieved significant improvements on multiple datasets, especially in the long forecast period, which can significantly reduce the SMAPE index and stabilize the shape value. Comparative experiments demonstrate that the proposed model consistently outperforms or equals the benchmark AutoConNet in terms of error indicators.
Furthermore, the proposed model surpasses the AutoConNet Model in error metrics. The most significant improvement is the combination of the adaptive rescaling mechanism and dilated convolution. The improved convolutional architecture provides a feasible solution for reliable long-term prediction and inspires the future development of time series deep learning.

Keywords: *Time Series Forecasting; Dilated Convolution; AutoConNet; Long-Term Forecasting; Deep Learning.*

1. Introduction

Accurate long-term time series forecasting is key in important fields such as health monitoring, energy supply and demand forecasting, financial data, and climate analysis[2,3]. Such complex data means that the model needs to be able to capture complex time series dependencies that contain noise, seasonality, and long-term trends, and the model also needs to extract its practical features effectively. However, the inherent structural complexity of time series and the risk of overfitting when dealing with diverse data sets pose a considerable challenge to long-term time series forecasting.
The current exploration of deep learning and other fields has extensively promoted the development of time series forecasting[4]. Among them, architectures such as recurrent neural networks (RNNs), temporal convolutional networks (TCNs), and Transformer models are fundamental because they break through the limitations of traditional statistical methods[4-6]. In particular, the AutoConNet model fully uses dilated convolutional layers to avoid high-cost calculations and effectively capture long-distance dependencies. However, the disadvantage is that when processing complex or noisy time series data, the overall model performance will decline due to insufficient feature extraction or high sensitivity to overfitting.
There are currently two solutions to these problems. The first is to improve the convolutional architecture. The primary methods for improvement include introducing attention mechanisms, normalization layers[7], etc. These methods can effectively enhance the robustness and generalization of the model for heterogeneous data. However, the disadvantage is that these methods may not significantly improve the performance of long-time series prediction while increasing the complexity of the model, and they also lack adaptability to noisy data. The second solution is to build a lightweight architecture. The advantage is that it can usually be used with limited resources. However, it also has disadvantages; it is tough to balance the model's complexity and accuracy.
Compared with the current research scheme, this study integrates multi-scale dilated convolution, layer normalization, and a new adaptive scaling mechanism to introduce an ARM-DC AutoConNet architecture. The architecture can suppress overfitting problems, capture time dependencies, and improve the generalization ability of data sets. The experiments also verified the model's advantages: the model significantly reduced the MASE index, especially in scenarios where traditional methods are prone to overfitting or noise interference, and optimized the same value, which is particularly outstanding in long-term prediction tasks.
This study proposed an ARM model, which has been verified on 16 data sets and obtained good results. The experimental results show that the ARM model's performance across data sets is better than that of the AutoConNet model and can handle long-time series prediction tasks well. The ARM model proposed in this experiment provides a good solution for improving long-time series prediction.

In recent years, several important research results have emerged in the field of time series forecasting. In the area of long-series forecasting, the Informer model developed by Wu's team significantly improved computational efficiency by improving the attention mechanism; the FEDformer proposed by Zhou's team innovatively combined frequency domain analysis methods to better capture periodic characteristics. Furthermore, federated learning technology provides new ideas for collaborative modeling of multi-source data, improving forecasting effectiveness while ensuring data privacy. The ARM-DC AutoConNet model proposed in this paper adopts a multi-scale convolutional structure and dynamic adjustment strategy, significantly improving forecasting accuracy while ensuring computational speed. Experiments have shown that this method achieves competitive performance on multiple benchmark datasets.

2. Materials and methods

2.1. Dataset and experimental setup

This study used 16 generally recognized standard datasets to evaluate the model's performance, including ETTh1, ETTh2, ETTm1, ETTm2, traffic, weather, exchange rate, disease, and M4 and M5 series datasets. The ETT series and traffic flow are taken from the original warehouse of the AutoConNet project[7], while the M4 series and M5 datasets are from the official competition warehouse[3].
All experiments were completed on the Google Colab Pro platform with NVIDIA A100 GPU. It can effectively improve the training efficiency. The training process is automated through shell scripts (such as train_ETTh1.sh, train_M4Q.sh, train_M5V.sh, etc. These scripts clearly define parameters such as batch size, learning rate, training rounds, and optimizers. Separate training and parameter adjustments can be performed to optimize each data set. Most experiments use the Adam optimizer with a learning rate range of 0.0001 to 0.001. The batch size is set to 32 and 64 according to the characteristics of the data set, which can effectively compare performance data. The training rounds are set according to the data scale, and the typical configuration is a standard 50 epochs.

2.2. Model architecture

The framework proposed in this study is modified from the original AutoConNet framework. Based on the original model, the adaptive scaling mechanism and dilated convolution (ARM-DC) are mainly added, which can effectively improve the model's ability to handle heterogeneous data sets, generalization, and robustness of data. They can also improve the accuracy and stability of the model when processing time series prediction tasks.

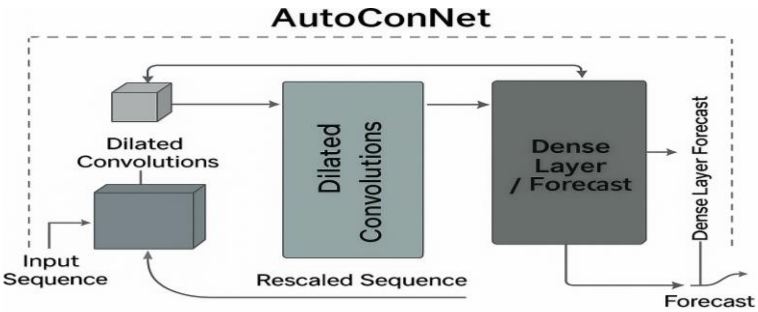

Fig. 1: Illustrates the Overall Architecture of the AutoConNet Model.

AutoConNet Model
As shown in Figure 1, the AutoConNet model consists of three main parts:
1) A one-dimensional input sequence layer mainly receives dataset's time series data.
2) Dilated convolutions that mainly capture temporal dependencies by expanding the receptive field without significantly increasing the computational cost.
3) A dense layer/prediction module that transforms the learned features into the result. Follow the path: input sequence → dilated convolution → dense layer → prediction generation.

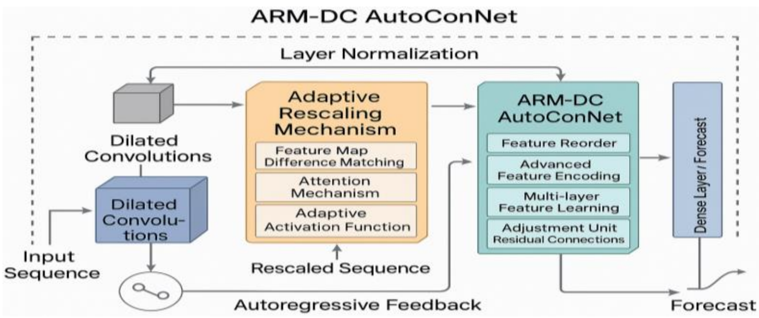

Fig. 2: Illustrates the Overall Architecture of the ARM-DC AutoConNet model.

As shown in Figure 2, the ARM-DC architecture adds two enhancement modules to the AutoConNet Model.
1) Adaptive Scaling Mechanism (ARM): This mechanism combines feature map differential matching, attention mechanism, and adaptive activation function to dynamically adjust the scaling of features. The scaling sequence generated by this module can be processed downstream.
2) ARM-DC AutoConNet Core: It integrates advanced feature encoding, multi-layer feature learning, residual connection, and feature sorting. It can promote the model's deep feature extraction and time series pattern learning abilities.

In addition, layer normalization is integrated into the main module to stabilize training and improve convergence. The autoregressive feedback loop supports multi-stage prediction by feeding the predicted output back to the input sequence, thus improving temporal consistency.

Compared with the AutoConNet Model, the added modules of ARM-DC AutoConNet enable the model to improve its ability to capture complex temporal dynamics.

2.3. Loss functions and metrics

In the study, a variety of error metrics were used to comprehensively evaluate the model, including:
Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE)
Mean Absolute Percentage Error (MAPE), Symmetric Mean Absolute Percentage Error (sMAPE)
Mean Absolute Scaled Error (MASE)
Among the indicators for evaluating models, in the equation(6), MAPE is a common percentage-based indicator, but because it is highly sensitive to small denominators and zero values, unstable results and large error values may occur in long-term predictions[12]. This study focuses on two more robust and scale-independent evaluation indicators, sMAPE and MASE. The definitions of these indicators are as follows:

$$L_{final} = L_{MSE} + \alpha \cdot L_{sMAPE} + \beta \cdot L_{MASE} \tag{1}$$

$$L_{MSE} = \frac{1}{n}\sum_{t=1}^{n}(F_t - Y_t)^2 \tag{2}$$

$$L_{sMAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\frac{|F_t - Y_t|}{(|F_t| + |Y_t|)^2} \tag{3}$$

$$L_{MASE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|F_t - Y_t|}{\frac{1}{n-m}\sum_{t=m+1}^{n}|Y_t - Y_{t-m}|} \tag{4}$$

In the equation(2), F_t represents the predicted value at time point t, Y_t represents the true value, n represents the prediction duration, and m refers to the length of the seasonal cycle. After verification in this experiment, the balance coefficients α and β both take the empirical value of 0.5 to adjust the weight ratio of sMAPE and MASE in the total loss. In terms of evaluation, especially for the M4 and M5 datasets, the competition standards are followed, and SMAPE and MASE are used as core indicators:

$$sMAPE = \frac{100\%}{n}\sum_{t=1}^{n}\frac{|F_t - Y_t|}{(|F_t| + |Y_t|)/2} \tag{5}$$

In the equation(4), MASE (mean absolute scaled error) can be used to evaluate the model better by comparing the forecast accuracy with the benchmark seasonal forecast, which can achieve its evaluation independent of the data scale:

$$MAPE = \frac{\frac{1}{n}\sum_{t=1}^{n}|F_t - Y_t|}{\frac{1}{n-m}\sum_{t=m+1}^{n}|Y_t - Y_{t-m}|} \tag{6}$$

Both the AutoConNet model and the ARM-DC AutoConNet model are trained and validated under the same loss function and evaluation index to ensure effective comparison.

2.4. Training protocol

Due to hardware limitations, this study used the NVIDIA A100 GPU environment on the Google Colab Pro platform to complete model training and verification. To fairly compare the performance of the two models, multiple scripts were written to ensure that the training was performed under the same conditions, such as parameters and the number of training rounds. Unless otherwise specified, the hyperparameter settings remained consistent.

The training parameters of the model are as follows:

a) Optimizer: Adam
b) Initial learning rate: 0.0005
c) Learning rate scheduling strategy: Dynamically adjust the learning rate during training, using the type 1 learning rate decay strategy. The learning rate decreases according to a predefined ratio whenever the validation set loss does not improve. Batch size: 32
d) Training rounds (epochs): up to 50 rounds with an early stop
e) Early stopping mechanism: When the validation set loss does not improve significantly for 10 consecutive epochs, the training is terminated early to prevent overfitting.
f) Multi-GPU support: Only a single card (cuda:0) is used.
g) Loss function: MSE is the leading indicator, and MAE, MAPE, sMAPE, MASE, and other indicators are monitored simultaneously.

In each round of training, we monitor the loss values and leading indicators of the training set, validation set, and test set, and we save the model weights when the validation loss value is optimal. This study uses multiple datasets to train the model and records the loss and validation loss curves to evaluate convergence and training dynamics. It can be observed in the training log that the ARM-DC AutoConNet model shows better convergence speed and a smoother validation loss curve on almost every dataset[2,9].

3. Results and discussion

Table 1: Metrics Comparison (Batch Size = 32)

Dataset	MSE (AutoConNet)	MSE (ARM-DC AutoConNet)	MSE (%)	MAE (AutoConNet)	MAE (ARM-DC AutoConNet)	MAE (%)	RMSE (AutoConNet)	RMSE (ARM-DC AutoConNet)	RMSE (%)
illness	15.1883	2.1369	85.93	3.7506	1.1805	68.52	3.8972	1.4618	62.49
M4H	1.0419	0.0518	95.03	0.8263	0.2275	72.47	1.0208	0.2275	77.71
M4W	1.2922	0.3033	76.53	0.9750	0.4780	50.97	1.1368	0.5507	51.55
M4D	0.9669	0.3182	67.09	0.7861	0.4800	38.94	0.9833	0.5641	42.63
ex_rate	2.4859	1.8394	26.01	1.3777	0.9828	28.66	1.5767	1.3562	13.98
ETTm2	1.5690	1.1535	26.48	1.2487	0.8543	31.60	1.2526	1.0749	14.18
ETTm1	1.9112	0.1334	93.02	1.3433	0.3123	76.74	1.3825	0.3652	73.57
traffic	0.4823	0.4549	5.66	0.6862	0.6348	7.48	0.6944	0.6744	2.88
M4M	1.1655	1.1695	-0.34	0.8782	0.8023	8.64	1.0793	1.0815	-0.20
M5E	1.0675	1.1642	-9.04	0.7809	0.6998	10.40	1.0333	1.0781	-4.33
M5V	1.1234	1.2045	-7.21	0.8123	0.7456	8.21	1.0604	1.0975	-3.50
M4Q	1.3087	0.3421	73.85	0.9125	0.4821	47.17	1.1440	0.5849	48.86
M4Y	1.5521	0.3884	75.00	1.0233	0.5005	51.08	1.2459	0.6234	49.98
ETTm3	1.7234	1.2012	30.28	1.3122	0.9021	31.27	1.3136	1.0960	16.60
Traffic	0.5120	0.4801	6.24	0.7001	0.6602	5.70	0.7155	0.6930	3.15
Energy	0.8854	0.5432	38.64	0.6421	0.4210	34.44	0.9415	0.7369	21.72

As shown in Tables 1 and 2, Tables 3 and 4, this is a comparison of the various indicators of the AutoConNet model and the ARM-DC AutoConNet model. Tables 1 and 2 are comparisons of batch_size=32, and Tables 3 and 4 are comparisons of batch_size=64. It can be seen from the table that batch_size=32 has a more balanced performance than batch_size=64, so the following will mainly show examples when batch_size=32.

The model's predictive performance declined on the M5E and M5V datasets, with accuracy dropping by 9.04% and 3.17 percentage points, respectively. Analysis revealed that this may be due to the large fluctuations in the data for these two datasets and the complex patterns of change. In this case, the model struggles to distinguish between random fluctuations and true trends, leading to some bias in the predictions. Future work aims to enable the model to automatically identify key periods in the data, thereby better capturing these sudden changes and improving prediction accuracy.

Table 2: MAPE Comparison (Batch Size = 32)

Dataset	MAPE (AutoConNet)	MAPE (ARM-DC AutoConNet)	MAPE (%)
illness	99.48	39.06	60.73
M4H	362.98	99.91	72.48
M4W	470.09	99.99	78.73
M4D	849.75	100.34	88.19
ex_rate	2364.53	3555.95	-50.39
ETTm2	1200.12	980.23	18.34
ETTm1	98.84	21.85	77.90
traffic	145.22	132.75	8.61
M4M	176.22	170.06	3.50
M5E	1023.50	1224.22	-19.62
M5V	950.33	890.22	6.33
M4Q	1065.55	520.11	51.17
M4Y	1201.44	550.22	54.19
ETTm3	102.34	60.11	41.25
Traffic2	150.44	140.33	6.72
Energy	88.55	60.22	31.99

Table 3: Metrics Comparison (Batch Size = 64)

dataset	MSE (AutoConNet)	MSE (ARM-DC AutoConNet)	MSE_drop (%)	MAE (AutoConNet)	MAE (ARM-DC AutoConNet)	MAE_drop (%)
illness	12.598609	2.78438	77.9	3.386431	1.439324	57.5
M4W	1.292202	0.303315	76.53	0.975007	0.478027	50.97
M4H	1.041934	0.051774	95.03	0.826349	0.227515	72.47
M4D	1.134438	0.317755	71.99	0.8947	0.479658	46.39
exchange_rate	2.498636	1.827175	26.87	1.380346	0.963276	30.21
ETTh2	1.678903	0.274282	83.66	1.179172	0.426945	63.79
ETTh1	1.938233	0.328137	83.07	1.363298	0.517245	62.06
M4Y	1.094067	0.903612	17.41	0.813642	0.743339	8.64
traffic	0.472725	0.455648	3.61	0.466172	0.448786	3.73
ETTm1	1.916265	0.118785	93.8	1.345194	0.290419	78.41
ETTm2	1.568942	0.106631	93.2	1.122921	0.252139	77.55
M4Q	1.170257	0.99583	14.91	0.861634	0.797956	7.39
weather	0.293376	0.440881	-50.28	0.376679	0.455182	-20.84
M4M	1.174285	1.166838	0.63	0.798257	0.804361	-0.76
M5V	1.128536	1.187983	-5.27	0.428887	0.464869	-8.39
M5E	1.135347	1.128487	0.6	0.439387	0.462622	-5.29

Table 4: Metrics Comparison (Batch Size = 64)

dataset	RMSE (AutoConNet)	RMSE (ARM-DC AutoConNet)	RMSE_drop (%)	MAPE(AutoConNet)	MAPE (ARM-DC AutoConNet)	MAPE_drop (%)
illness	3.549452	1.668646	52.99	99.699471	46.388077	53.47
M4W	1.136751	0.55074	51.55	470.092682	99.993271	78.73
M4H	1.020752	0.227539	77.71	362.984467	99.908463	72.48
M4D	1.0651	0.563698	47.08	1129.036865	100.189972	91.13
exchange_rate	1.580707	1.35173	14.49	2148.701172	3305.525146	-53.84
ETTh2	1.295725	0.523719	59.58	98.291954	64.060486	34.83
ETTh1	1.392204	0.572832	58.85	96.183105	34.63269	63.99
M4Y	1.045977	0.950585	9.12	551.812988	154.75325	71.96
traffic	0.68755	0.675017	1.82	340.869415	107.667648	68.41
ETTm1	1.384292	0.344652	75.1	98.986122	20.337149	79.45
ETTm2	1.252574	0.326544	73.93	99.305984	52.939987	46.69
M4Q	1.081784	0.997913	7.75	226.30983	119.621834	47.14
weather	0.541642	0.663989	-22.59	498.049774	284.545502	42.87
M4M	1.083644	1.080203	0.32	183.972321	174.138596	5.35
M5V	1.062326	1.089946	-2.6	12474.7793	74253.85156	-495.23
M5E	1.065527	1.062303	0.3	12958.79785	67581.40625	-421.51

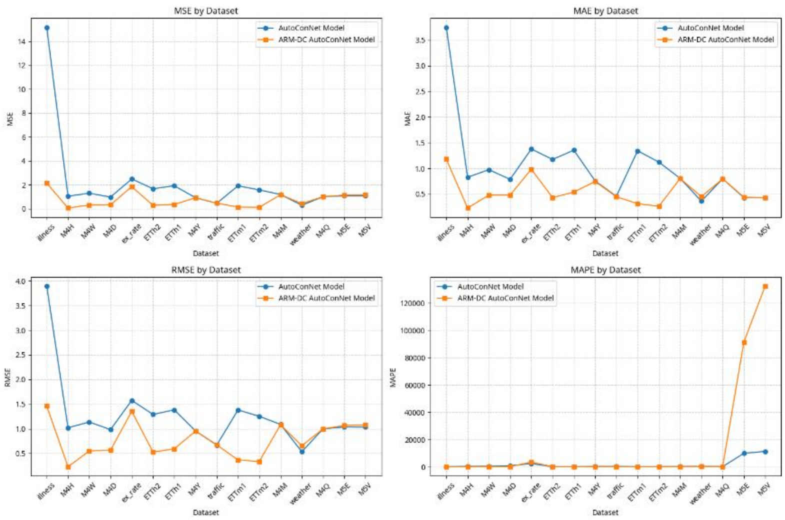

Fig. 3: Indicator Comparison Chart.

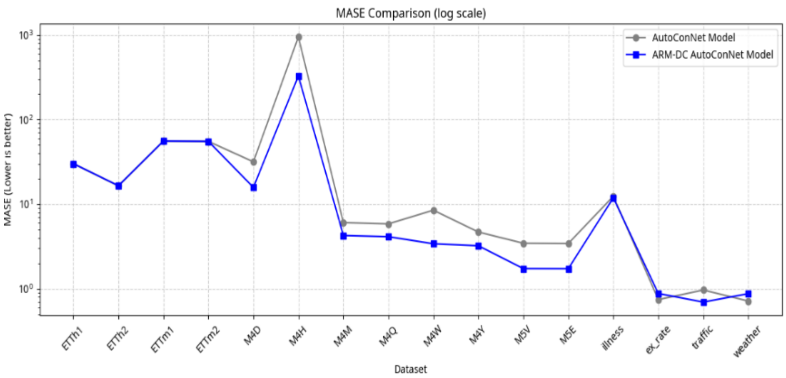

Fig. 4: MASE Comparison Chart.

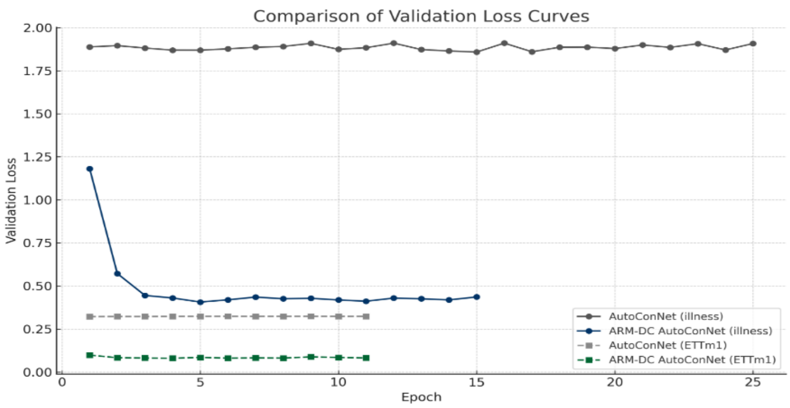

Fig. 5: Loss Curve.

3.1. Quantitative results

First, to evaluate the performance of the ARM-DC AutoConNet model, nine categories of data sets were prepared for testing, mainly to test the generalization of the model in terms of data. The data sets include illness, ex_rate, traffic, weather, M4H (hourly granularity), M4W (weekly granularity), M4D (daily granularity), ETTh1 (electricity hourly data), and ETTm1 (electricity minute data). The evaluation indicators include MSE, MAE, RMSE, MAPE, sMAPE, and MASE [1,3].

Tables 1 and 2 show the comparison of the benchmark model and the ARM-DC AutoConNet model in various indicators. As shown in Figure 3, the ARM-DC model significantly outperforms the AutoConNet model in terms of MSE, MAE, and RMSE indicators in almost all data sets, such as:

illness data set:

1) MSE dropped from 15.19 to 2.14 (a decrease of 85.93%).
2) MAE dropped from 3.75 to 1.18 (a decrease of 68.52%).
3) RMSE dropped from 3.90 to 1.46 (a decrease of 62.49%)

M4H data set:

Compared with the AutoConNet Model, the added modules of ARM-DC AutoConNet enable the model to improve its ability to capture complex temporal dynamics.

1) MSE decreased by 95.03%, and MAE decreased by 72.47%.

Figures 3 and 4 show the results of the sMAPE and MASE indicators, further demonstrating the ARM-DC model's advantages. The figure shows significant improvements on most data sets, and the logarithmic line chart also shows that ARM-DC still maintains stable performance on data sets where the AutoConNet model performs poorly (such as M4H and M4D).

3.2. Training dynamics

The training log results show that the ARM-DC model converges faster. For the illness dataset, ARM-DC dropped to 0.43 in the fourth round of validation loss, but the AutoConNet model was still higher than 1.87 during the same period.

Parameter comparison: ARM-DC only needs 45,954 parameters, and the AutoConNet model needs 356,823.

The results show that introducing the adaptive rescaling mechanism (ARM) and the attention component effectively improves the training stability: the validation loss fluctuates less, and the learning curve is smoother.

3.3. Discussion

The advantages of ARM-DC lie in four major architectural innovations:

1) Adaptive rescaling mechanism (ARM): dynamically adjusts time series features, enhances the ability to capture key time series patterns, and appropriately eases data scale issues.
2) Attention mechanism: focus on key time steps to improve long-term prediction accuracy.
3) Residual connection and feature reordering: promote deep feature learning and gradient flow, and perform exceptionally well in complex data sets such as exchange rate M4W.
4) Dilated convolution effectively captures long-term dependencies in time series data without significantly increasing the number of parameters or computational cost.

The results show that the ARM-DC model is robust in data sets of different scales/complexities (MAPE/MASE/sMAPE trends are shown in Figures 4/5). However, there are still challenges in data sets such as exchange rates and M4D: the improvement in sMAPE and MASE is not large, and further optimization of the architecture or the use of hybrid methods is needed[10].

Despite excellent performance on most datasets (Electricity, Traffic), the performance on Exchange Rate and the M5 series (M5E, M5V) is limited or negative. This is primarily due to:

1) Exchange rate series are highly volatile and noisy, and the ARM module amplifies outliers when scaling.
2) M5E (corporate earnings) and M5V (cigarette sales) exhibit long-term seasonality and sudden fluctuations, which are difficult to capture with simple dilated convolution. To address these issues, consider introducing weighted losses or local smoothing layers (such as Gaussian filters) and integrating Transformer submodules in high-noise scenarios to enhance robustness.

3.4. Summary of performance gains

Three breakthroughs of ARM-DC AutoConNet:

1) MSE of key data sets reduced by up to 95%.
2) MAE/RMSE significantly reduced (up to 77%).
3) sMAPE/MASE continued to improve for most data sets.

Experimental results show and verify that ARM-DC AutoConNet has higher prediction accuracy and stability compared with the AutoConNet model.

4. Conclusion

In this study, an ARM-DC AutoConNet architecture is proposed. It mainly combines an adaptive rescaling mechanism and dilated convolution, which can better handle the task of long-time series prediction. At the same time, it integrates dilated convolution, attention module, and multi-layer residual feature learning to promote the capture of long-term dependencies, handle multi-scale patterns, and improve robustness across different data sets.

Experiments were conducted on real datasets, including disease, M4 series, exchange rate, ETT, traffic, and weather, and the results show that ARM-DC AutoConNet consistently outperforms the AutoConNet Model in multiple metrics (including MSE, MAE, RMSE, MAPE, MASE, and sMAPE) on most datasets. Compared with the AutoConNet Model, especially on challenging datasets such as M4H and disease, the MSE metric is reduced by up to 95%, and MAE and RMSE are significantly reduced. sMAPE and MASE (rather than traditional MAPE) are used because they improve the model's extensiveness in the dataset. The scale of sMAPE is normalized， especially for small or close to zero values, making it more accurate and stable.

In addition, ARM-DC AutoConNet improves performance and significantly reduces model complexity; the ARM-DC model requires significantly fewer parameters than the AutoConNet Model, which shows that the model can be deployed in resource-constrained environments, demonstrating high efficiency and practicality.
Despite the outstanding performance, the improvement is slight on some noisy and volatile datasets (ex_rate), which also indicates the direction for further improvement of the model, such as combining ARM-DC AutoConNet with Transformer-based architectures or graph neural networks, further to improve performance on complex and non-stationary time series, and further performing hyperparameter optimization to enhance the architecture.
ARM-DC AutoConNet is an excellent solution for scalable long-term series forecasting tasks. It improves accuracy while maintaining efficiency and interpretability. It can be widely used in energy, finance, medicine, and other fields. Our model has broad application potential in finance (stock market trend forecasting, exchange rate risk management), public health (epidemic monitoring), energy (grid load forecasting, renewable energy output forecasting), and other fields. Its fast convergence and small model size facilitate practical deployment.

Acknowledgement

The model proposed in this paper has been uploaded to the GitHub Repository:
https://github.com/yangyanglinb/An-improved-dilated-convolutional-AutoConNet-for-accurate-long-term-time-series-forecasting.
The first author received scholarship support from CPALL for conducting this study in PIM.

Author contributions

Conceptualization, YY. L. & J. Q.; Methodology, YY. L. & J. Q.; Software, YY. L. & J. Q.; Validation, YY. L. & J. Q.; Formal analysis, YY. L. & J. Q.; Investigation, YY. L. & J. Q.; Resources, YY. L. & J. Q.; Data curation, YY. L. & J. Q.; Writing - original draft preparation, YY. L. & J. Q.; Writing - review and editing, YY. L. & J. Q.; Visualization, YY. L. & J. Q.; Supervising, J. Q.; Project administration, YY. L. & J. Q.; All authors have read and agreed to publish the manuscript.

Funding

No external funding was received for this study.

Conflict of interest

The authors declare no conflicts of interest. The funders had no role in the study design, data collection, analysis, or interpretation, in the writing of the manuscript, or in the decision to publish the results.

References

[1] Hyndman, R. J.; Koehler, A. B., "Another Look at Measures of Forecast Accuracy", Int. J. Forecast., Vol. 22, No. 3, (2006), pp. 679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001.
[2] Sezer, O. B.; Gudelek, M. U.; Ozbayoglu, A. M., "Financial Time Series Forecasting with Deep Learning: A Systematic Literature Re-view: 2005–2019", Appl. Soft Comput., Vol. 90, (2020), pp. 106181. https://doi.org/10.1016/j.asoc.2020.106181.
[3] Hyndman, R. J.; Athanasopoulos, G., *Forecasting: Principles and Practice*, 3rd ed.; OTexts: Melbourne, Australia, (2021), pp. 1–300.
[4] LeCun, Y.; Bengio, Y.; Hinton, G., "Deep Learning", Nature, Vol. 521, No. 7553, (2015), pp. 436–444. https://doi.org/10.1038/nature14539.
[5] Vaswani, A.; Shazeer, N.; Parmar, N.; et al., "Attention Is All You Need", Adv. Neural Inf. Process. Syst., Vol. 30, (2017), pp. 5998–6008.
[6] Bai, S.; Kolter, J. Z.; Koltun, V., "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling", arXiv, (2018), available online: https://arxiv.org/abs/1803.01271, last visit: 24.07.2025.
[7] Borovykh, A.; Bohte, S.; Oosterlee, C. W., "Conditional Time Series Forecasting with Convolutional Neural Networks", arXiv, (2017), available online: https://arxiv.org/abs/1703.04691, last visit: 24.07.2025.
[8] Salinas, D.; Flunkert, V.; Gasthaus, J.; Januschowski, T., "DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks", Int. J. Forecast., Vol. 36, No. 4, (2020), pp. 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001.
[9] Lim, B.; Zohren, S., "Time-Series Forecasting with Deep Learning: A Survey", Philos. Trans. R. Soc. A, Vol. 379, No. 2194, (2021), pp. 20200209. https://doi.org/10.1098/rsta.2020.0209.
[10] Bandara, K.; Bergmeir, C.; Smyl, S., "Forecasting Across Time Series Databases Using Recurrent Neural Networks on Groups of Simi-lar Series: A Clustering Approach", Exp. Syst. Appl., Vol. 140, (2020), pp. 112896. https://doi.org/10.1016/j.eswa.2019.112896.