Deep Residual MLP Networks: Enhancing Precision and Reliability in Energy Forecasting

Abstract

Despite the long-standing success of Multilayer Perceptrons (MLPs) across diverse applications, increasing their depth often introduces overfitting and gradient degradation. To overcome these limitations, this paper proposes a novel hybrid architecture that synergistically integrates MLPs with Residual Networks (ResNet). Specifically, MLPs serve as nonlinear mapping functions within ResNet blocks, while skip connections preserve gradient flow to enable stable training in deeper networks. The model is optimized using the Adam algorithm for its rapid convergence and further enhanced through systematic hyperparameter tuning via grid search. Comprehensive evaluations are performed on three critical energy forecasting domains: electricity demand, petroleum products, and renewable energy generation, with comparisons against 10 state-of-the-art models. The proposed framework demonstrates superior predictive accuracy, achieving a mean absolute percentage error (MAPE) of 4.495% in petroleum demand forecasting, significantly outperforming all baseline methods. These results underscore the model's robustness and practical relevance for real-world energy forecasting applications.

Keywords: Residual Network, Multilayer Perceptron, Adaptive Moment Estimation, Gridsearch.

1 Introduction

In recent decades, the evolution of deep learning techniques has revolutionized the field of artificial intelligence, paving the way for sophisticated models capable of capturing complex patterns in data. Among these, Multilayer Perceptrons (MLPs) have played a foundational role, serving as the building blocks for various neural network architectures.

A layered network of perceptrons is first introduced by Frank Rosenblatt in his book *Perceptron* [1] [2] [3]. The perceptrons in his book is composed of an input layer, a hidden layer with randomized weights which did not learn, and an output layer with learning connections. But this is seen as a extreme learning machine [4] but not a deep learning network. Although this early form of MLP was not considered a deep

*Corresponding author: yushuxiang@mails.swust.edu.cn

learning network, it laid the groundwork for subsequent advancements. In 1965, Alexey Grigorevich Ivakhnenko and Valentin Lapa published the first deep-learning feedforward network, known as the Group Method of Data Handling, which did not yet utilize stochastic gradient descent [5] [6]. Two years later, Shun'ichi Amari introduced a deep-learning network capable of classifying non-linearly separable pattern classes, marking the first use of stochastic gradient descent in such networks [7]. And his team also built a five-layered feedforward network, demonstrating the feasibility of deep learning architectures. The modern backpropagation method, a crucial component of MLP training, was first published in 1970 by Seppo Linnainmaa [8]. This efficient application of a chain-rule-based supervised learning approach revolutionized the training of neural networks by enabling the propagation of errors through the network to update the model parameters. Subsequent improvements to the backpropagation algorithm, including its standardization by Paul Werbos in 1982 [9], and experimental analyses conducted by David E. Rumelhart et al. in 1985 [10], further solidified its importance in the field of deep learning.

So far, The MLP model and its variants have been widely used in many fields , such as finance [11], bioinformatics [12], transportation [13], agricultur [14], medical [15] and *etc.*. The MLP especially plays an important role in time series analysis including regression and classification. In 2023, FINANNISA ZHAFIRA and *etc.* combine LSTM and MLP to establish a model that can effectively reduce training costs [16]. In the same year, Si-An Chen and *etc.* present Time-Series Mixer (TSMixer) which is a novel architecture designed by stacking multi-layer perceptrons (MLPs) and prove its surperior performance on a real-world retail dataset [17]. Sujan Ghimire and *etc.* propose an novel hybrid method which integrates convolutional neural network (CNN) with MLP and forecasts global solar radiation (GSR) successfully [18].

However, the depth of the MLP model is limited by the vanishing gradient problem, making it difficult to train deeper networks. To solve the problem of vanishing gradient, in 1991, Sepp Hochreiter introduced skip connections or residual connections in the long short-term memory (LSTM) recurrent neural network to solve this problem [19]. Subsequently, in 2015, the concept of Highway Networks was proposed, applying the concept of forget gates in LSTM to the feedforward neural network, allowing information to spread in the network and alleviating the vanishing gradient problem [20]. Then, based on Highway Networks, ResNet further simplifies the structure, removes forget gates, and uses simple skip connections directly, so that signals can be propagated directly without the intervention of the gating mechanism. This structure has proven to be very effective in training very deep neural networks [21].

Building upon the aforementioned research, this paper presents an innovative integration of ResNet with MLP for forecasting across three distinct types of energy data. Specifically, the MLP is employed as the mapping function within the ResNet architecture. To further optimize model performance, we develop a dual-phase optimization strategy combining Adam optimizer with GridSearchCV, which significantly enhances the predictive capability of our model. The main contributions of this work can be summarized as follows:

- This work propose a hybrid MLP-ResNet framework that utilizes MLP as residual mapping functions, effectively combining skip connections with original inputs. This innovative architecture maintains MLP's powerful feature extraction capabilities while inherently addressing gradient-related challenges, thereby significantly enhancing model generalization.

- Our approach integrates the Adam optimization algorithm for adaptive learning rate adjustment with systematic hyperparameter optimization via GridSearchCV. This dual optimization strategy ensures both computational efficiency and model performance maximization.

- Through comprehensive experiments across three distinct energy domains - equipment monitoring (transformer oil temperature), energy trade (crude oil imports), and renewable generation (wind power) - and comparison with 10 state-of-the-art baseline models, this work demonstrate the superior performance and robust generalizability of our approach.

In the rest of the paper, the theory of MLP-ResNet and its solution will be shown in Section 2; applications in 3 real-world cases in energy field will be represented in Section 3; the conclusion of this paper is shown in Secton 4.

2 Theoretical Framework

2.1 Knowledge Background

2.1.1 Multiple Layer Perceptron (MLP)

Formally, an MLP consists of an input layer, one or more hidden layers, and an output layer. Each layer is composed of numerous artificial neurons, also referred to as perceptrons or nodes, interconnected via weighted connections [22]. The primary function of the MLP is to transform input data through successive layers of nonlinear transformations, ultimately producing an output prediction. A simple mlp network structure with one-hidden-layer is shown in Fig.(1).

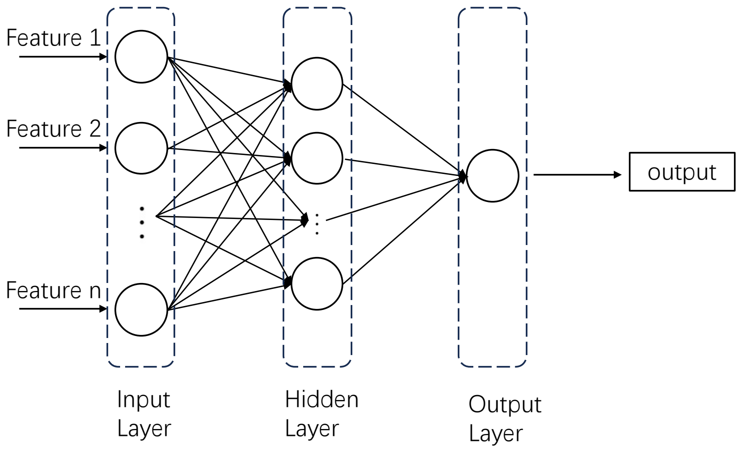

Fig. 1. Simple mlp network structure (single hidden layer)

Mathematically, the forward propagation process of an MLP can be expressed as follows: for each layer l, the output $\mathbf{x}^{(l)}$ is computed as the application of a nonlinear activation function σ to the linear transformation of the previous layer's output $\mathbf{x}^{(l-1)}$, incorporating weights $\mathbf{W}^{(l)}$ and biases $\mathbf{b}^{(l)}$:

$$\begin{aligned} \mathbf{z}^{(l)} &= \mathbf{W}^{(l)}\mathbf{x}^{(l-1)} + \mathbf{b}^{(l)} \\ \mathbf{x}^{(l)} &= \sigma\left(\mathbf{z}^{(l)}\right) \end{aligned} \tag{1}$$

During training, the parameters (weights and biases) of the MLP are optimized to minimize a predefined loss function, typically through backpropagation and gradient-based optimization techniques. Backpropagation involves the systematic calculation of gradients with respect to the parameters of the network, facilitating parameter updates in the direction that reduces the loss [23].

MLPs are characterized by their universal approximation capabilities, enabling them to approximate arbitrary functions with sufficient capacity and data. However, their effectiveness is contingent upon various factors, including network architecture design, activation functions, optimization algorithms, and hyperparameter tuning.

2.1.2 Residual Network (ResNet)

Recent advancements in the field of image recognition have underscored the critical role of network depth, particularly in Convolutional Neural Networks (CNNs), as elucidated by recent studies [24]. However, the efficacy of deeper networks is marred by a phenomenon termed degradation, wherein the accuracy of the model plateaus and subsequently declines rapidly with increasing depth. Notably, this degradation does not stem from overfitting but rather from optimization challenges.

Addressing this inherent limitation, ResNet (Residual Network) presents a pioneering solution by introducing a residual learning framework. Unlike conventional CNNs where each layer aims to directly learn the underlying target function $H(x)$ [25], ResNet adopts a distinctive learning objective defined as $F(x) := H(x) - x$. This formulation epitomizes residual learning, where the network endeavors to learn the residual information of x in $H(x)$. Distinguishing between the architectural setups of conventional CNN blocks and ResNet blocks is shown in Fig.(2). By reframing the learning task in terms of residual functions, ResNet facilitates more efficient optimization, as it is inherently easier to learn residuals than to directly learn complex target functions. This approach enables ResNet to navigate around the degradation issue by traversing a detour through residual learning pathways [26].

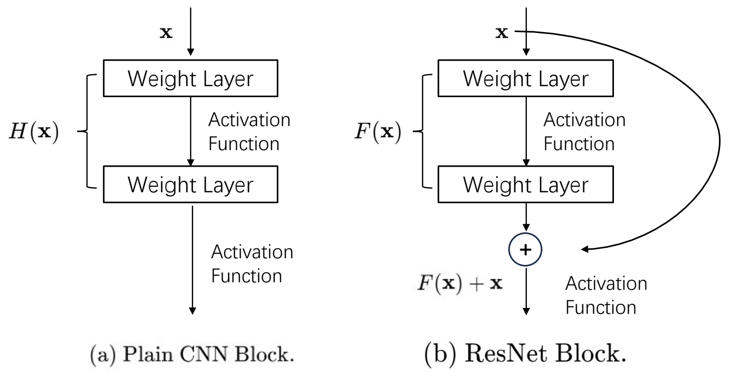

Fig. 2. Distinguishing between the architectural setups of conventional CNN blocks and ResNet blocks.

The core architectural feature of ResNet is the incorporation of "shortcut connections" or "skip connections," which facilitate identity mapping. Through these connections, the original input x is added directly to the output of the stacked layers, thereby enabling the flow of information without significant alteration. This mechanism not only fosters smoother gradient flow during backpropagation but also mitigates the vanishing gradient problem commonly encountered in deep networks [27].

2.2 The proposed MLP-ResNet model

2.2.1 The representation of MLP-ResNet model and its solution

As we mentioned before, it's apparent that the most existing machine learning models including MLP often face degradation phenomenon which means the accuracy of the model declines rapidly when increasing depth. In order to better forecast and enhance the MLP's versatility, we propose the model which combines the MLP and the ResNet which is called MLP-ResNet (MLPRS) in this paper. The structure of the MLPRS is shown in Fig.(3) and the ◯ with number means the neuron index in each layer. The number of MLPRS BLOCK depends on the depth of MLPRS.

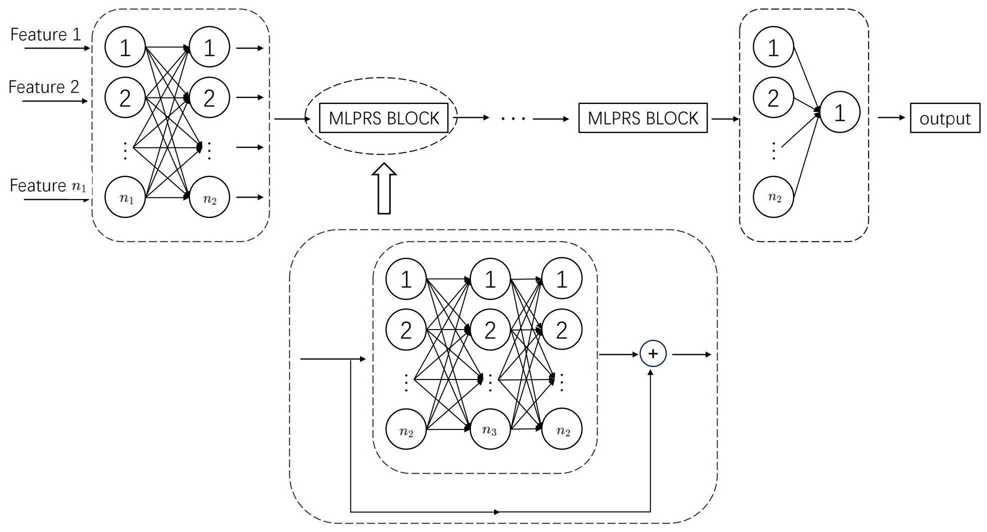

Fig. 3. Structure of MLPRS

Suppose (X, y) is the input data of the model which X has n_1 features, it is composed of $(X_t, y_t)(t = 1, 2, \cdots, n)$ and we constuct the MLPRS model with k depth. The MLP in the proposed model has one-hidden-layer, we could obtain the output of MLPRS as follows:

$$h_0 = W^{(1)}X + b^{(1)}, \tag{2}$$

where h_0 is the output of ResNet's input layer.

When it come to the first MLPRS BLOCK, we could write the mathematical expression as follows:

$$\begin{aligned} h_1 &= W^{(2)}h_0 + b^{(2)}, \\ h_2 &= \delta(h_1), \\ h_3 &= W^{(3)}h_2 + b^{(3)}, \\ h_4 &= W_1^{(4)}h_3 + h_0, \end{aligned} \tag{3}$$

where h_1, h_2, h_3 are the state of the nerurons in MLP model (the input layer, hidden layer and output layer of MLP respectively), and h_4 is the output of the first block. $\delta(\cdot)$ is the activation function of the MLP model.

Similar to the first block, the formula of the second block could be written as follows:

$$\begin{aligned} h_5 &= W^{(2)}h_4 + b^{(2)}, \\ h_6 &= \delta(h_5), \\ h_7 &= W^{(3)}h_6 + b^{(3)}, \\ h_8 &= W_2^{(4)}h_7 + h_4, \end{aligned} \tag{4}$$

After the iteration of k depth, the k-th block's output h_{4k} could be obtained:

$$h_{4k} = W_k^{(4)}h_{4n-1} + h_{4n-4}. \tag{5}$$

Finally, we could calculate the output $\hat{y}$ of the ResNet:

$$\hat{y} = W^{(5)}h_{4n} + b^{(5)} \tag{6}$$

In Eq.(2)(3)(4)(5)(6), $W^{(i)}(i = 1, 2, 3, 4, 5)$ and $b^{(j)}(j = 1, 2, 3, 5)$ are the parameters of neural network in MLPRS. $W_p^{(4)}(p = 1, 2, \cdots, k)$ is the k-th element in $W^{(4)}$.The specific shape of the parameter matrix are shown in Tab.1.

Table 1: The specific shape of the parameter matrix.

parameter	$W^{(1)}$	$W^{(2)}$	$W^{(3)}$	$W^{(4)}$	$W^{(5)}$
shape	(n_1, n_2)	(n_2, n_3)	(n_3, n_2)	$(1, k)$	$(n_2, 1)$
parameter	$b^{(1)}$	$b^{(2)}$	$b^{(3)}$	$b^{(5)}$	
shape	$(1, n_2)$	$(1, n_3)$	$(1, n_2)$	$(1, 1)$	

The activation function has a variety of choices which depends on specific application scenarios, here we use Tanh as the activation function of MLPRS. The output range of the Tanh function is within $[-1, 1]$, which helps to reduce the problem of vanishing gradients and helps the network converge. The Tanh function $\delta(\cdot)$ could be express as follows:

$$\delta(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{7}$$

2.2.2 Adam algorithm for training the MLP-ResNet model

Normally, the neural network could not obtain an analytical solution, and the proposed MLPRS model in this paper is no exception. So we need to use optimization algorithms to get its solution such as Gradient Descent [28], Stochastic Gradient Descent [29], Adam [30]. In this paper, we introduce the Adam algorithm to train the proposed model due to its efficiency, robustness, and adaptability.

First, we need to define the training error e_t at each point (X_t, y_t):

$$e_t = y_t - (W^{(5)} h_{4n} + b^{(5)}), \tag{8}$$

Thus, we could obtain the sum of training error:

$$E(\boldsymbol{W}, \boldsymbol{b}) = \frac{1}{n} \sum_{t=1}^{n} e_t^2 = e^T e, \tag{9}$$

where $\boldsymbol{W}$ is composed of $W^{(i)}$ and $\boldsymbol{b}$ is composed of $b^{(j)}$. $\boldsymbol{W}$ and $\boldsymbol{b}$ are the parameters whcih need to be solved by Adam.

Then, in order to complete Adam, we need to finish the gradient descent. So we have to get the gradient of $E(\boldsymbol{W}, \boldsymbol{b})$:

$$\boldsymbol{L} = [\frac{\partial E}{\partial \boldsymbol{W}}, \frac{\partial E}{\partial \boldsymbol{b}}], \tag{10}$$

where $\boldsymbol{L}$ is the gradient.

Different with ordinary gradient descent, it introduces the concept of the modified bias-corrected first moment estimate $\hat{m}_t$ and bias-corrected second raw moment estimate $\hat{v}_t$ to speed up convergence, their mathematical expressions are as follows:

$$m_t = \mu_1 \cdot m_{t-1} + (1 - \mu_1) \cdot \boldsymbol{L}, \tag{11}$$

$$v_t = \mu_2 \cdot v_{t-1} + (1 - \mu_2) \cdot \boldsymbol{L}^2, \tag{12}$$

$$\hat{m}_t = \frac{m_t}{1 - \mu_1^t}, \tag{13}$$

$$\hat{v}_t = \frac{v_t}{1 - \mu_2^t}, \tag{14}$$

where μ_1 and μ_2 mean decay rates which are used to control the decay speed of the first and second moments of the gradients, respectively.

Finally, we can obtain the iterative formula:

$$\begin{bmatrix} \boldsymbol{W}^{t+1} \\ \boldsymbol{b}^{t+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}^{t} \\ \boldsymbol{b}^{t} \end{bmatrix} - l_1 \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}. \tag{15}$$

where l_1 is the learning rate of Adam and ϵ is a small constant. The complete algorithm of Adam is shown in Algorithm 1:

Algorithm 1: The complete process of Adam training MLPRS

```
  Input: $E(\boldsymbol{W}, \boldsymbol{b})$ (Eq.(9)), Learning rate $l_1$, max_epochs
  Initialize: $[\boldsymbol{W}, \boldsymbol{b}] \leftarrow random()$;
  $\mu_1 \leftarrow 0.9$; $\mu_2 \leftarrow 0.999$;
  $m_0 \leftarrow 0$; $v_0 \leftarrow 0$;
  $epoch \leftarrow 0$;
1 while $epoch < max\_epochs$ do
2     ephoch = ephoch + 1 ;
3     $\boldsymbol{L} \leftarrow$ Eq.(10);
4     $\widehat{m}_t, \widehat{v}_t \leftarrow$ Eq.(13)(14);
5     $\begin{bmatrix} \boldsymbol{W}^{t+1} \\ \boldsymbol{b}^{t+1} \end{bmatrix} \leftarrow$ Eq.(15);
6 end
7 return $[\boldsymbol{W}, \boldsymbol{b}]$
```

2.2.3 Optimal Model Parameter Selection Using Gridsearch algorithm

In Section 2.2.2, we obtain the parameter set $[\boldsymbol{W}, \boldsymbol{b}]$ through Adam algorithm, but the depth k, learning rate l_2 and the number of nunber of neurons $n_r (r = 2, 3)$ are still need to be tuned. Here we introduce the Gridsearch algorithm to tune the parameters.

The basic principle of GridSearch is to exhaustively search all possible parameter combinations in the parameter space, then perform cross-validation on each parameter combination, and select the parameter combination with the best performance [31].

Suppose we have a model parameter space Θ, where each parameter combination can be represented by a vector $\boldsymbol{\theta}$ which consisting of k, l_2, n_2, n_3. Our goal is to find the best parameter combination $\boldsymbol{\theta}^*$ given the training data set D_{train}, so that the model can perform better on the validation data set D_{val} for optimal performance. Its mathematical principle can be expressed as follows [32]:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \Theta} f(\boldsymbol{\theta}, D_{\text{train}}, D_{\text{val}}), \tag{16}$$

where the term $f(\boldsymbol{\theta}, D_{\text{train}}, D_{\text{val}})$ represents the performance metric obtained by training the model with parameter combination $\boldsymbol{\theta}$ on the validation set D_{val}. Here we use Negative mean square error (NMSE) to calculate the metric:

$$f(\boldsymbol{\theta}, D_{\text{train}}, D_{\text{val}}) = -\frac{1}{|D_{\text{val}}|} \sum_{i \in D_{\text{val}}} (y_i - \hat{y}_i)^2 \tag{17}$$

3 Applications

To verify the applicability and stability of the proposed model, we use 3-real-world data in energy field. The data set has important practical significance and it will be discussed further in later context. Moreover, min-max mapping is used to prevent overflow here. The MAPE metric are used to measure the performance of the model and its mathematical expression is as follows:

$$\frac{1}{s}\sum_{k\in U}\frac{|\hat{y}(t)-y(t)|}{|y(t)|} \tag{18}$$

where U is the training or testing set and s is the length of U.

10 models are used to compare, and the information is shown in Tab.2.

Table 2: Information of models used for comparison

Full Name	Abbreviation	Reference	Year
Gated Recurrent Unit	gru	[33]	2017
Random Forest Regression	rf	[34]	2001
Extreme Gradient Boosting	xgb	[35]	2015
Long Short-Term Memory	lstm	[36]	2000
Support Vector Regression	svr	[37]	1996
Convolution Neural Network	cnn	[38]	2015
Multilayer Perceptron	mlp	[39]	2009
CNN-LSTM	cnnlstm	[40]	2019
Convolutional LSTM	convlstm	[41]	2017
General Regression Neural Network	grnn	[42]	2004

3.1 Case 1:Electricity Transformer Oil Temperature

In power distribution problems, the accuracy of voltage distribution is crucial [43]. The distribution of electricity needs to be adjusted according to the needs of different regions, and this adjustment often depends on the continuous use of electricity. However, predicting future demand in a specific region is a difficult task as it is affected by various factors such as working days, holidays, seasons, weather, temperature, etc. Any incorrect prediction may damage the operation of the electrical transformer. As a result, there is currently no very effective way to predict future electricity usage, and managers are forced to make decisions based on empirical numbers, which are often higher than actual demand. This results in unnecessary waste and depreciation of electricity and equipment.

The transformer oil temperature can reflect the operating status of the electrical transformer, so in this article we use oil temperature prediction to better solve the voltage distribution problem to avoid unnecessary waste [44].

In this paper, we collect hourly transformer oil temperature data from 1:00 on June 1, 2018 to 19:00 on June 26, 2018 from the website *https://github.com/zhouhaoyi/ETDataset*. The first 496 points are used to train and the rest 124 points are used to test. The training plot is shown in Fig.4 and the testing plot is shown in Fig.5. The MAPE value of all the models are presented in Table 3.

As shown in Fig. 4, most models demonstrate strong performance for in-sample predictions, with differences among them not readily discernible from the comparison plots.

However, a contrasting pattern emerges on the test set, as illustrated in Fig. 5. Specifically, the MAPE results presented in Table 3 indicate that the MLP-ResNet model achieves the best performance, with a MAPE of 5.132%, outperforming the majority of competing models. In contrast, the traditional MLP exhibits the poorest performance, recording a MAPE of 7.058%, which further confirms the degradation issues discussed earlier and highlights the superiority of the MLP-ResNet architecture. Furthermore, the Random Forest (RF) model attains the lowest MAPE of 1.620% on the training set but fails to sustain this performance on the test set, where the MAPE increases to 6.57%, suggesting potential overfitting. These results underscore the significant advantages of incorporating residual structures in improving model generalization and stability, thereby providing robust technical support for energy forecasting applications.

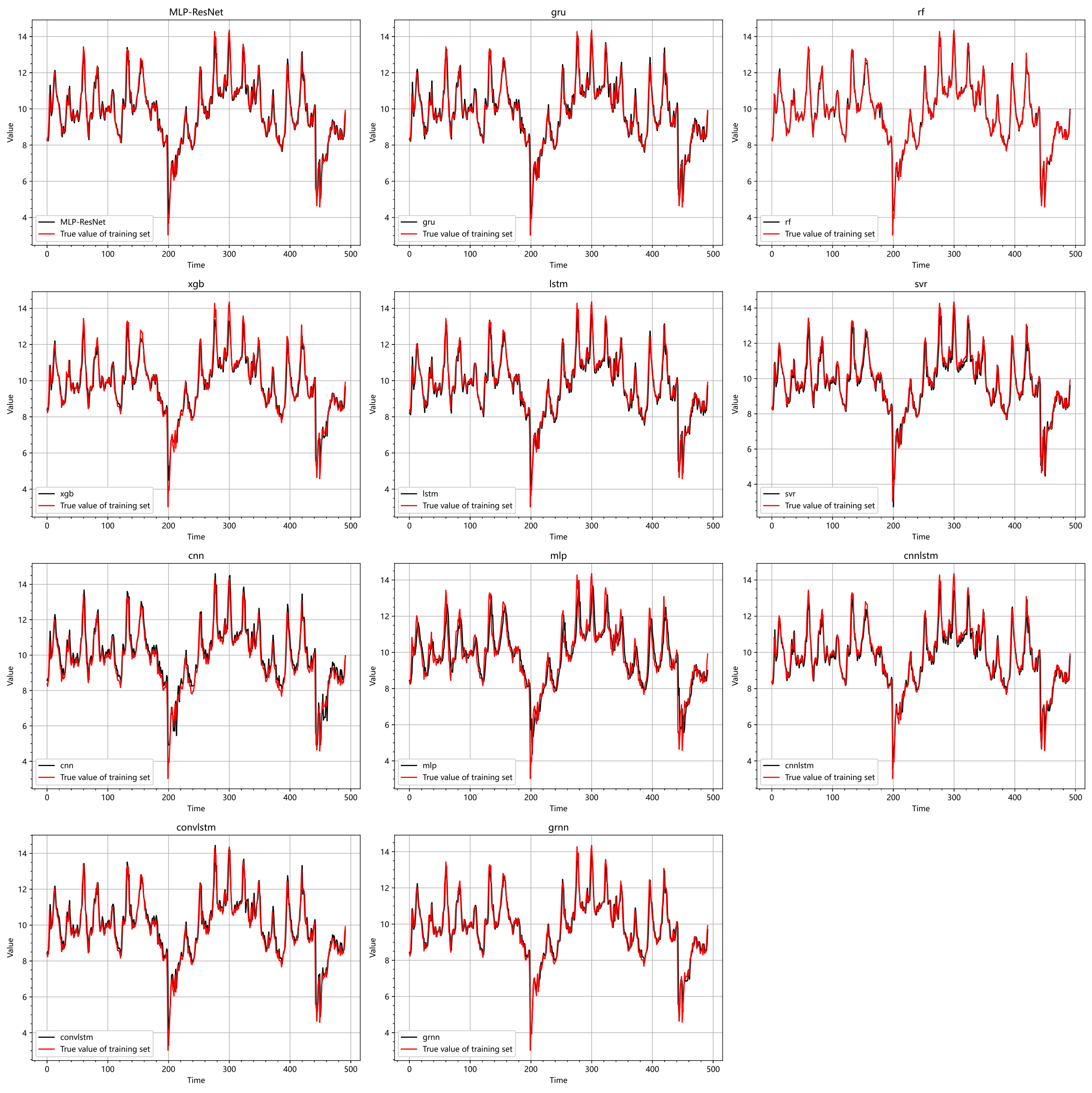

Fig. 4. Prediction values of Electricity Transformer Oil Temperature training set

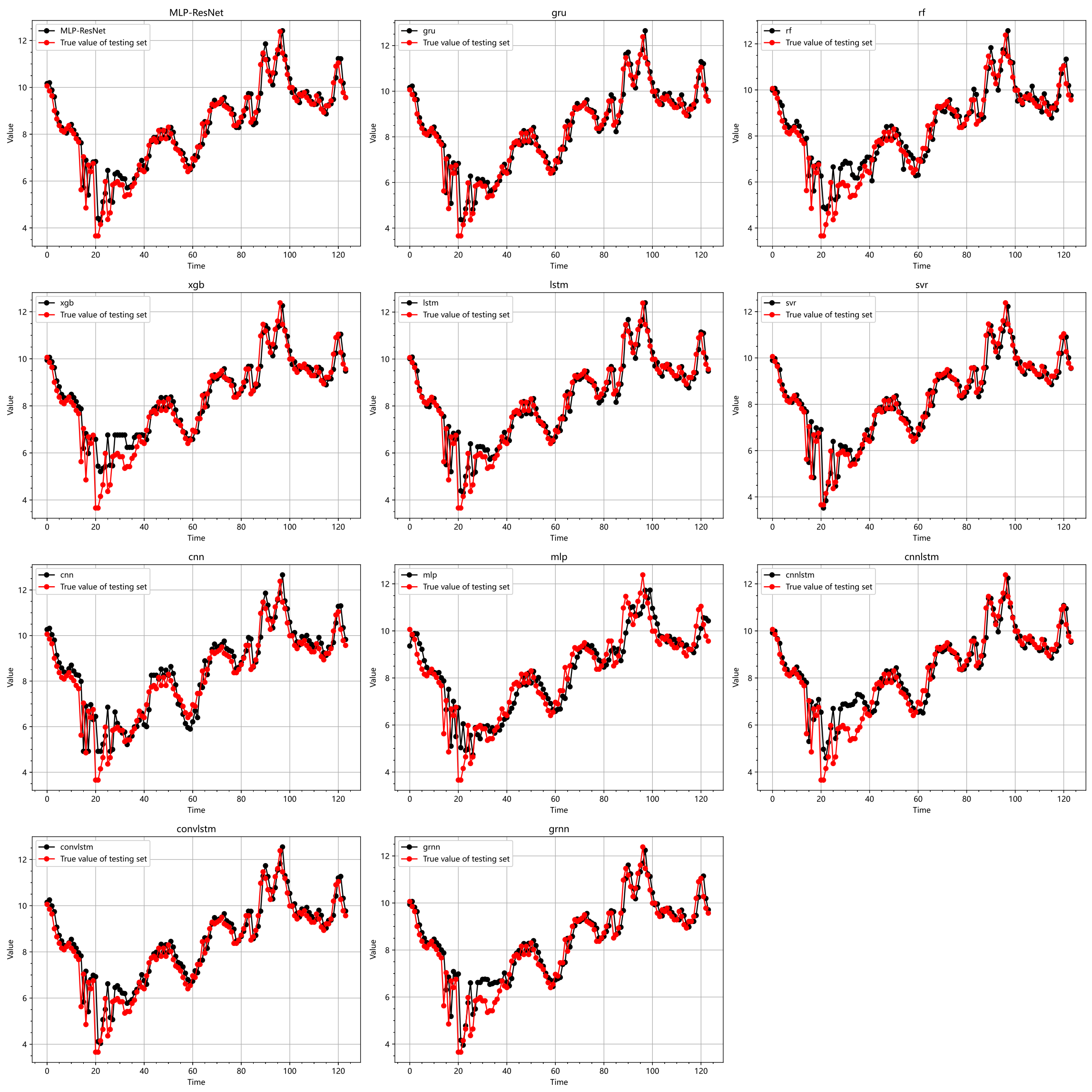

Fig. 5. Prediction values of Electricity Transformer Oil Temperature testing set

Table 3: MAPE for training and testing of all the models in Case-1

Model	**MLP-ResNet**	**gru**	**rf**	**xgb**	**lstm**	**svr**	**cnn**	**mlp**	**cnnlstm**	**convlstm**	**grnn**
Train	3.623	3.684	**1.620**	3.509	3.800	4.038	4.190	5.744	3.649	3.695	3.221
Test	**5.132**	5.155	6.570	5.966	5.240	5.217	6.609	7.058	6.440	5.538	5.862

3.2 Case 2:U.S. Imports of Crude Oil and Petroleum Products

Petroleum and its products are a key component of the global economy and have profound impacts on energy markets, trade balances and geopolitics [45]. Therefore, accurate forecasts and analysis of U.S. crude oil and petroleum product imports are critical to the stability of international energy markets and the development of the global economy [46].

In this paper, we collect the monthly data from January 15, 1981 to June 15, 2023, and the data comes from the U.S. Energy Information Administration. For convenience of expression, we called the data as ICOP. The first 408 points are used for training and the last 102 points are used for testing. The training and testing plot are shown in Fig.6 and Fig.7. The MAPE value are shown in Tab.4.

Based on the performance observed on both in-sample and out-of-sample data, it is evident that the majority of models demonstrate excellent results, indicating their strong suitability for this dataset. Among them, the MLP-ResNet model leads with a MAPE of 4.495%. Notably, deep learning–based models such as MLP-ResNet, CNN, and LSTM exhibit significantly better results on the test set compared to the training set. For instance, the CNN model’ s MAPE reaches as high as 9.070% on the training set but remains at a much lower 4.709% on the test set. This clearly reflects the robust generalization capability of these neural network models, enabling them to maintain satisfactory performance even when training results appear suboptimal.

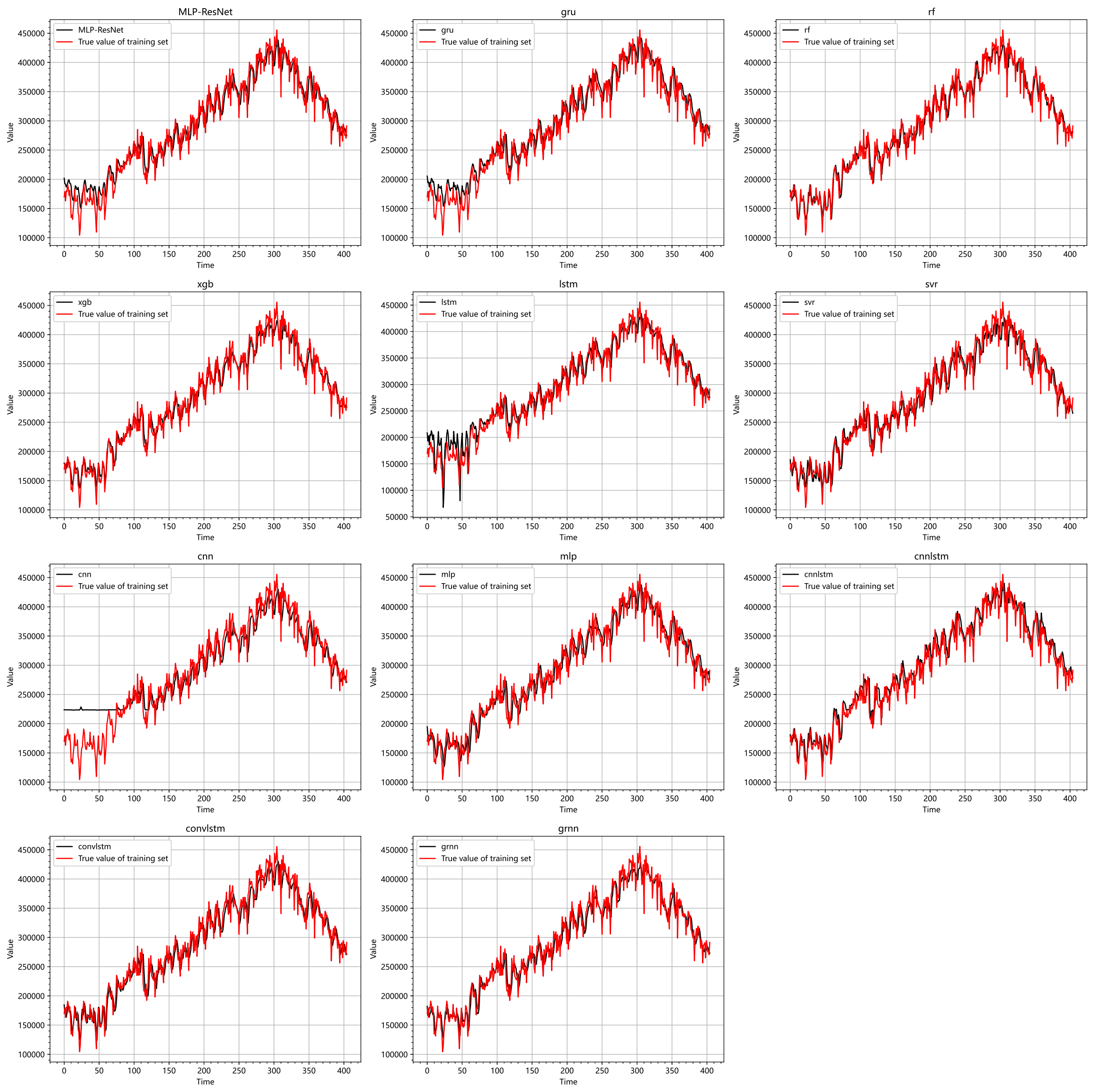

Fig. 6. Prediction values of ICOP training set

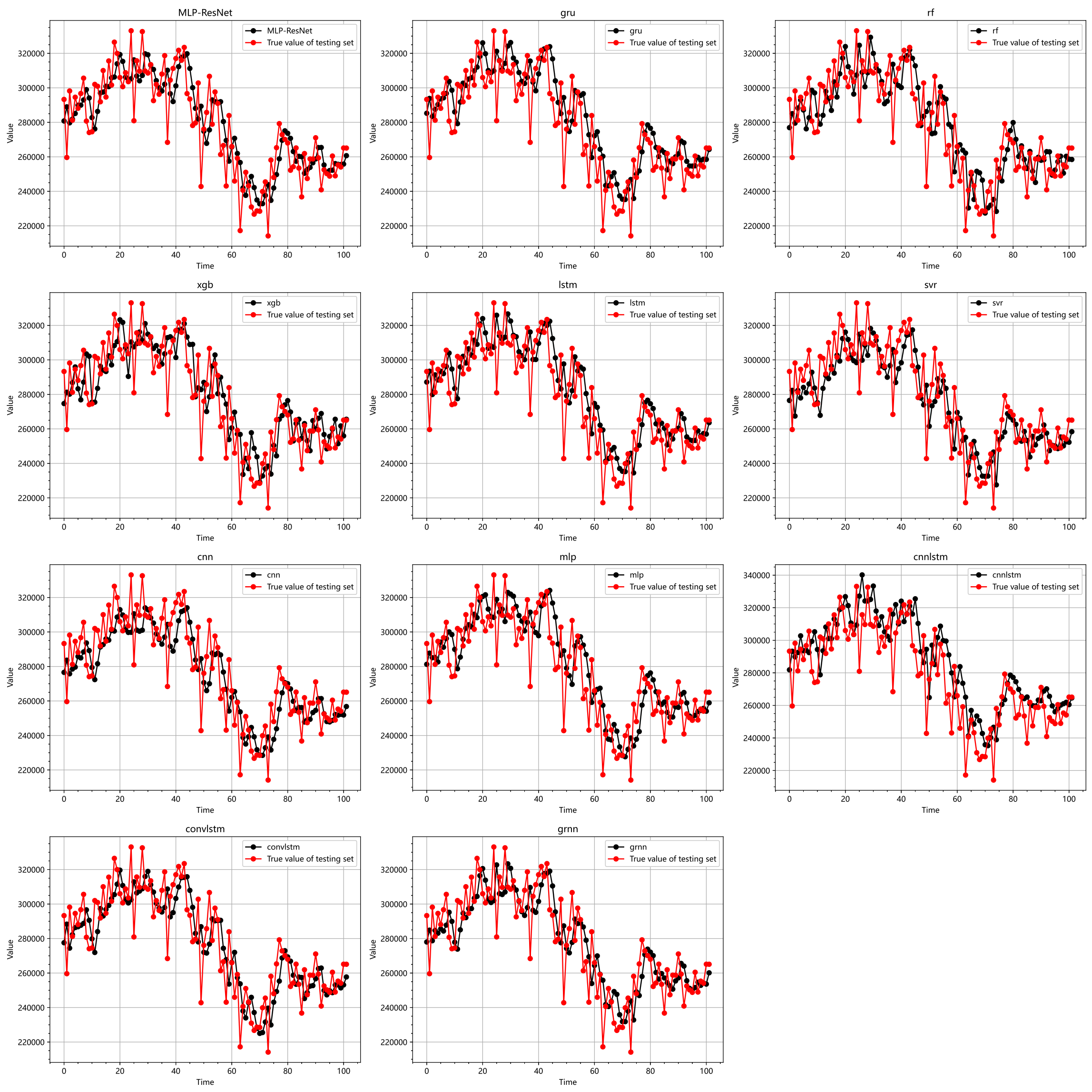

Fig. 7. Prediction values of ICOP testing set

Table 4: MAPE for training and testing of all the models in Case-2

Model	MLP-Resnet	gru	rf	xgb	lstm	svr	cnn	mlp	cnnlstm	convlstm	grnn
Train	6.144	6.169	3.842	**3.704**	6.516	5.705	9.070	5.941	5.074	5.453	4.922
Test	**4.495**	4.580	4.818	4.637	4.531	4.754	4.709	4.748	4.996	4.514	4.528

3.3 Case 3:Inland Wind Turbine Power Generation

Wind energy is of great significance to combat climate change, reduce carbon emissions, and achieve sustainable energy development. Forecasts of inland wind turbine power generation can unveil the potential and feasibility of wind power generation in the region [47]. Similarly, comprehending the power generation of individual turbines in inland wind farms aids in optimizing energy production and supply.

Here, we use this hourly Inland Wind Turbine Power Generation data (IWTPG) which is from September 11 to October 7, 2015, and the data is found from the website *https://zenodo.org/records/5516539*. We use the first 420 points to train and the last 105 points to test. The training and testing plot are shown in Fig.8 and Fig.9. The performance of the models are shown in Tab.5.

From the comparison plots and the final tabulated results, it is evident that this dataset poses significant challenges for the models, with most exhibiting poor predictive performance. This indicates that the models fail to capture deeper latent information within the data, suggesting the need for more samples or additional features to achieve better results. Nevertheless, the MLP-ResNet model remains the best-performing model, achieving a MAPE of 34.914%, while the GRU model performs the worst, with a MAPE of 59.740%. Additionally, the MLP-ResNetx model outperforms the traditional MLP and shows better results on the test set than on the training set, further demonstrating the superiority of the MLP-ResNet architecture.

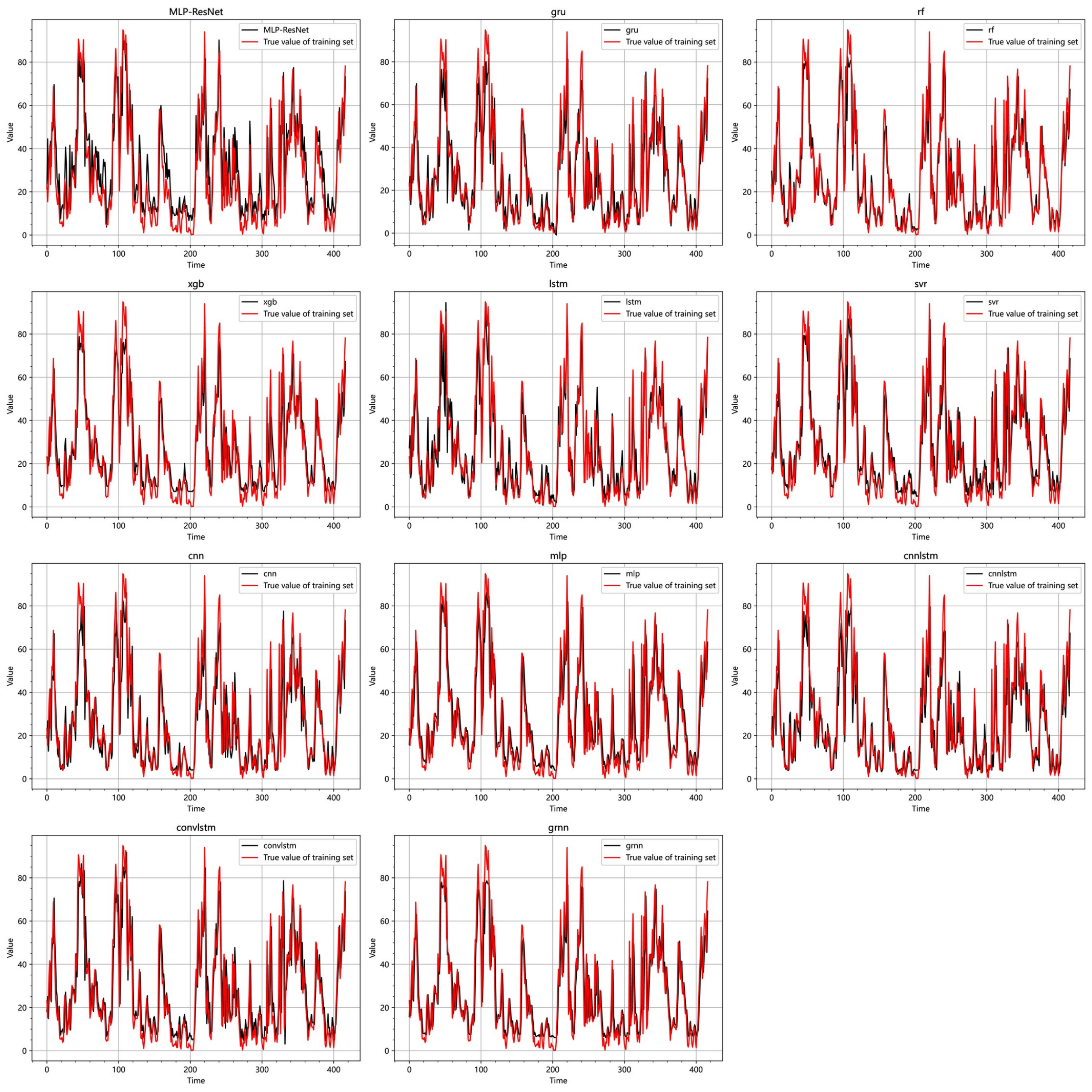

Fig. 8. Prediction values of IWTPG training set

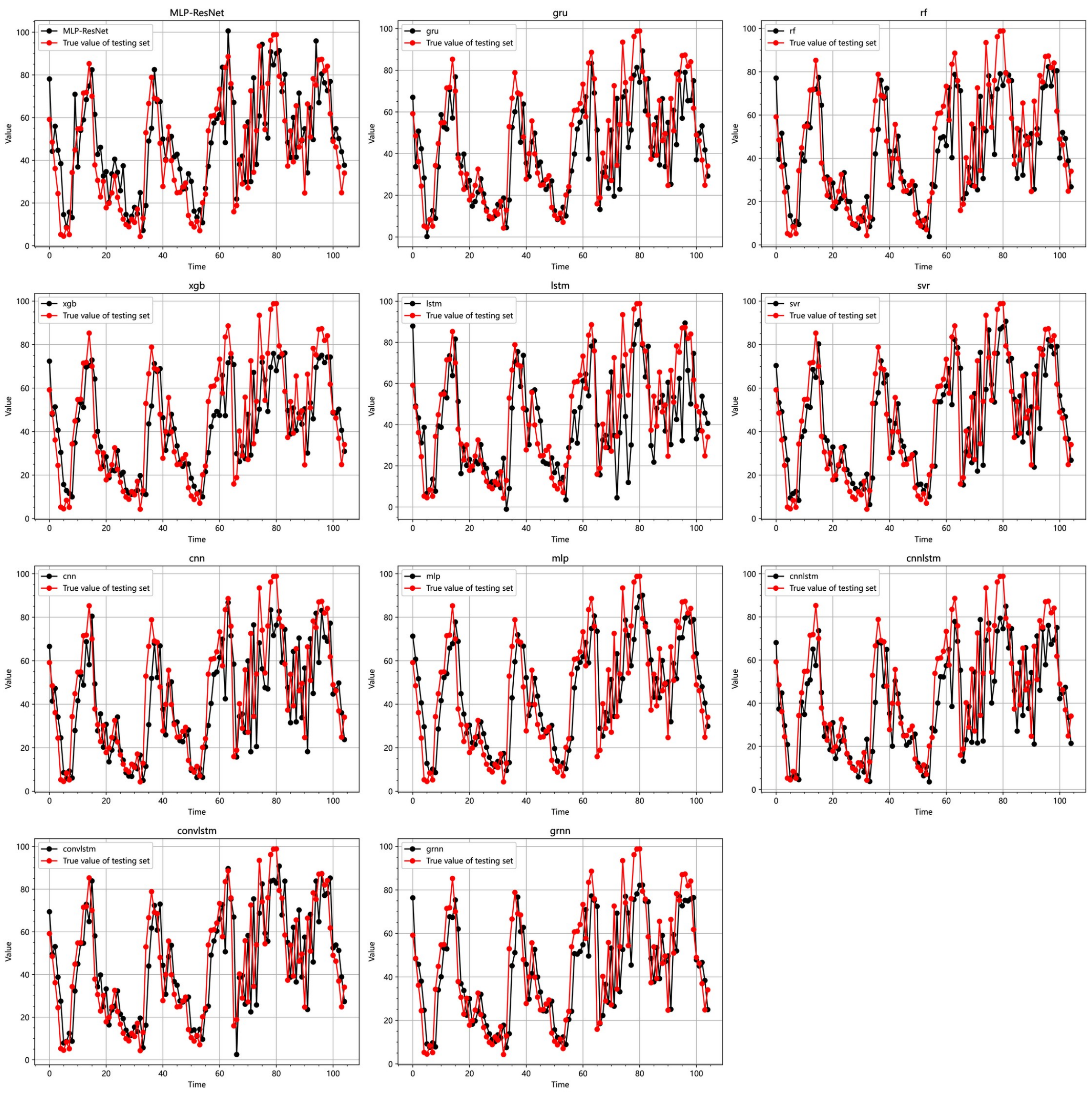

Fig. 9. Prediction values of IWTPG testing set

Table 5: MAPE for training and testing of all the models in Case-3

Model	MLP-ResNet	gru	rf	xgb	lstm	svr	cnn	mlp	cnnlstm	convlstm	grnn
Train	36.594	42.239	**28.035**	33.121	33.447	37.276	41.900	38.910	43.282	36.758	37.357
Test	**34.914**	59.740	44.011	38.140	69.743	39.190	50.164	36.553	53.151	44.628	39.438

3.4 Disscusion

Clearly, the proposed MLP-ResNet model performance on the training set is not outstanding, but it performs the best in testing in 3 cases and the prediction curve of MLP-ResNet is very close to the raw curve. This shows that it has strong generalization ability and prediction accuracy. This result shows the reliability and stability of the model in real scenarios. Meanwhile, the rf model achieves the best training MAPE value, but it frequently performs worse in testing. The reason for this may be that the model is not complex enough or overfitting occurs.

4 Conclusions

In this paper, we introduce the MLP-ResNet model, presenting a comprehensive theoretical framework, model training methodology, and parameter tuning approach. Moreover, according to 3 cases in Sec.3, it shows that the proposed model often performs best in testing and has good versatility. The way combines the ResNet and MLP could effectively improve the prediction accuracy and applicability. We believe that this kind of approach could have deeper research in the future.

References

[1] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386, 1958.

[2] Frank Rosenblatt. *Principles of Neurodynamics*. Spartan Press, 1962.

[3] Juergen Schmidhuber. Annotated history of modern ai and deep learning. *arXiv preprint arXiv:2212.11279*, 2022.

[4] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[5] Aleksei Grigorievich Ivakhnenko and Valentin Grigorievich Lapa. *Cybernetic Predicting Devices*. Naukova Dumka, 1965.

[6] Aleksei Grigorievich Ivakhnenko and Valentin Grigorievich Lapa. *Cybernetics and Forecasting Techniques*. Naukova Dumka, 1967.

[7] Shunichi Amari. A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, (3):299–307, 2006.

[8] Omar Hernández Rodríguez and Jorge M Lopez Fernandez. A semiotic reflection on the didactics of the chain rule. *The Mathematics Enthusiast*, 7(2):321–332, 2010.

[9] Paul J Werbos. Applications of advances in nonlinear sensitivity analysis. In *System Modeling and Optimization: Proceedings of the 10th IFIP Conference New York City, USA, August 31–September 4, 1981*, pages 762–770. Springer, 2005.

[10] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, Institute for Cognitive Science, University of California, San Diego, 1985.

[11] Junaid Maqbool, Preeti Aggarwal, Ravreet Kaur, Ajay Mittal, and Ishfaq Ali Ganaie. Stock prediction by integrating sentiment scores of financial news and mlp-regressor: A machine learning approach. *Procedia Computer Science*, 218:1067–1078, 2023.

[12] Yan Kang, Yulong Xu, Xinchao Wang, Bin Pu, Xuekun Yang, Yulong Rao, and Jianguo Chen. Hn-ppisp: a hybrid network based on mlp-mixer for protein–protein interaction site prediction. *Briefings in Bioinformatics*, 24(1):bbac480, 2023.

[13] Yujie Li, Zhiyun Yin, Yuchao Zheng, Huimin Lu, Tohru Kamiya, Yoshihisa Nakatoh, and Seiichi Serikawa. Pose estimation of point sets using residual mlp in intelligent transportation infrastructure. *IEEE Transactions on Intelligent Transportation Systems*, 24(11):13359–13369, 2023.

[14] Amirhossein Samii, Hojat Karami, Hamidreza Ghazvinian, Amirsaeed Safari, and Yashar Dadrasajirlou. Comparison of deep-lstm and mlp models in estimation of evaporation pan for arid regions. *Journal of Soft Computing in Civil Engineering*, 7(2):155–175, 2023.

[15] Shuo Gao, Wenhui Yang, Menglei Xu, Hao Zhang, Hong Yu, Airong Qian, and Wenjuan Zhang. U-mlp: Mlp-based ultralight refinement network for medical image segmentation. *Computers in Biology and Medicine*, 165:107460, 2023.

[16] Mostafa Abotaleb and Pushan Kumar Dutta. Optimizing long short-term memory networks for univariate time series forecasting: a comprehensive guide. *Hybrid Information Systems: Non-Linear Optimization Strategies with Artificial Intelligence*, 427, 2024.

[17] Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O Arik, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecasting. *arXiv preprint arXiv:2303.06053*, 2023.

[18] Sujan Ghimire, Thong Nguyen-Huy, Ramendra Prasad, Ravinesh C Deo, David Casillas-Pérez, Sancho Salcedo-Sanz, and Binayak Bhandari. Hybrid convolutional neural network-multilayer perceptron model for solar radiation prediction. *Cognitive Computation*, 15(2):645–671, 2023.

[19] Sepp Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. Master's thesis, Technische Universität München, 1991.

[20] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV 2016: European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[22] Talha Ansar and Waqar Muhammad Ashraf. Comparison of kolmogorov–arnold networks and multi-layer perceptron for modelling and optimisation analysis of energy systems. *Energy and AI*, 20:100473, 2025.

[23] James Spall, Xianxin Guo, and Alexander I Lvovsky. Training neural networks with end-to-end optical backpropagation. *Advanced Photonics*, 7(1):016004–016004, 2025.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[25] Keiron O'shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[26] Hyungeun Choi, Seunghyoung Ryu, and Hongseok Kim. Short-term load forecasting based on resnet and lstm. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6. IEEE, 2018.

[27] Heng Hua, Mingping Liu, Yuqin Li, Suhui Deng, and Qingnian Wang. An ensemble framework for short-term load forecasting based on parallel cnn and gru with improved resnet. *Electric Power Systems Research*, 216:109057, 2023.

[28] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *Advances in Neural Information Processing Systems*, 29, 2016.

[29] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993.

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] GSK Ranjan, Amar Kumar Verma, and Sudha Radhika. K-nearest neighbors and grid search cv based real time fault monitoring system for industries. In *2019 IEEE 5th international conference for convergence in technology (I2CT)*, pages 1–5. IEEE, 2019.

[32] Dwi Kartini, Dodon Turianto Nugrahadi, Andi Farmadi, et al. Hyperparameter tuning using gridsearchcv on the comparison of the activation function of the elm method to the classification of pneumonia in toddlers. In *2021 4th international conference of computer and informatics engineering (IC2IE)*, pages 390–395. IEEE, 2021.

[33] Rahul Dey and Fathi M Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pages 1597–1600. IEEE, 2017.

[34] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[35] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, and Tianyi Zhou. Xgboost: Extreme gradient boosting. R package version 0.4-2, 2015. https://cran.r-project.org/web/packages/xgboost.

[36] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural Computation*, 12(10):2451–2471, 2000.

[37] Harris Drucker, Christopher J Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. *Advances in Neural Information Processing Systems*, 9, 1996.

[38] Keiron O'shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[39] Marius-Constantin Popescu, Valentina E Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7):579–588, 2009.

[40] Tae-Young Kim and Sung-Bae Cho. Predicting residential energy consumption using cnn-lstm neural networks. *Energy*, 182:72–81, 2019.

[41] Seongchan Kim, Seungkyun Hong, Minsu Joh, and Sa-kwang Song. Deeprain: Convlstm network for precipitation prediction using multichannel radar data. *arXiv preprint arXiv:1711.02316*, 2017.

[42] Savita G Kulkarni, Amit Kumar Chaudhary, Somnath Nandi, Sanjeev S Tambe, and Bhaskar D Kulkarni. Modeling and monitoring of batch processes using principal component analysis (pca) assisted generalized regression neural networks (grnn). *Biochemical Engineering Journal*, 18(3):193–210, 2004.

[43] Suresh K Khator and Lawrence C Leung. Power distribution planning: A review of models and issues. *IEEE Transactions on Power Systems*, 12(3):1151–1159, 1997.

[44] TCBN Assunção, JL Silvino, and P Resende. Transformer top-oil temperature modeling and simulation. *International Journal of Mathematical and Computer Sciences*, 1:2, 2005.

[45] Onyenekenwa Cyprian Eneh. A review on petroleum: Source, uses, processing, products and the environment. *Journal of applied sciences*, 11(12):2084–2091, 2011.

[46] Perry Sadorsky. Modeling and forecasting petroleum futures volatility. *Energy Economics*, 28(4):467–488, 2006.

[47] Yun Wang, Runmin Zou, Fang Liu, Lingjun Zhang, and Qianyi Liu. A review of wind speed and wind power forecasting with deep neural networks. *Applied Energy*, 304:117766, 2021.