*Original Research Article*

A Residual Network based on MLP and its

2 applications in energy forecasting

3

# 5 Abstract

6 After decades of development, the Multilayer Perceptron (MLP) has made significant strides.

7 However, many MLP models still encounter the issue of degradation as depth increases. Fortunately,

8 recent research has shown that the Residual Network (ResNet) can mitigate or even overcome this

9 phenomenon. In this paper, we propose a Residual Network based on MLP to enhance model appli-

10 cability and stability, thereby improving data forecasting accuracy. We employ the Adam algorithm

11 for model training and the Gridsearch algorithm for hyperparameter tuning. In the Application

12 Section, we validate the effectiveness of our proposed model using three real-world energy-related

13 cases and compare it with 10 other machine learning models. Our results demonstrate that the

14 proposed model outperforms others in all three cases, highlighting its versatility and robustness.

15 **Keywords**: Residual Network, Multilayer Perceptron, Adaptive Moment Estima-

16 tion, Gridsearch.

# 17 1 Introduction

18 In recent decades, the evolution of deep learning techniques has revolutionized

19 the field of artificial intelligence, paving the way for sophisticated models capable of

20 capturing complex patterns in data. Among these, Multilayer Perceptrons (MLPs) have

21 played a foundational role, serving as the building blocks for various neural network

22 architectures.

23 A layered network of perceptrons is first introduced by Frank Rosenblatt in his

24 book *Perceptron* [[1](#_bookmark29)] [[2](#_bookmark30)] [[3](#_bookmark31)]. The perceptrons in his book is composed of an input layer,

25 a hidden layer with randomized weights which did not learn, and an output layer with

26 learning connections. But this is seen as a extreme learning machine [[4](#_bookmark32)] but not a

27 deep learning network. Although this early form of MLP was not considered a deep

28 learning network, it laid the groundwork for subsequent advancements. In 1965, Alexey

29 Grigorevich Ivakhnenko and Valentin Lapa published the first deep-learning feedforward

30 network, known as the Group Method of Data Handling, which did not yet utilize

31 stochastic gradient descent [[5](#_bookmark33)] [[6](#_bookmark34)]. Two years later, Shun’ichi Amari introduced a deep-

32 learning network capable of classifying non-linearly separable pattern classes, marking

33 the first use of stochastic gradient descent in such networks [[7](#_bookmark35)]. And his team also

34 built a five-layered feedforward network, demonstrating the feasibility of deep learning

35 architectures. The modern backpropagation method, a crucial component of MLP

36 training, was first published in 1970 by Seppo Linnainmaa [[8](#_bookmark36)]. This efficient application

37 of a chain-rule-based supervised learning approach revolutionized the training of neural

38 networks by enabling the propagation of errors through the network to update the model

39 parameters. Subsequent improvements to the backpropagation algorithm, including its

40 standardization by Paul Werbos in 1982 [[9](#_bookmark37)], and experimental analyses conducted by

41 David E. Rumelhart et al. in 1985 [[10](#_bookmark38)], further solidified its importance in the field of

42 deep learning.

43 So far, The MLP model and its variants have been widely used in many fields ,

44 such as finance [[11](#_bookmark39)], bioinformatics [[12](#_bookmark40)], transportation [[13](#_bookmark41)], agricultur [[14](#_bookmark42)], medical [[15](#_bookmark43)]

45 and *etc*.. The MLP especially plays an important role in time series analysis including

46 regression and classification. In 2023, FINANNISA ZHAFIRA and *etc*. combine LSTM

47 and MLP to establish a model that can effectively reduce training costs [[16](#_bookmark44)]. In the

48 same year, Si-An Chen and *etc*. present Time-Series Mixer (TSMixer) which is a novel

49 architecture designed by stacking multi-layer perceptrons (MLPs) and prove its surpe-

50 rior performance on a real-world retail dataset [[17](#_bookmark45)]. Sujan Ghimire and *etc*. propose an

51 novel hybrid method which integrates convolutional neural network (CNN) with MLP

52 and forecasts global solar radiation (GSR) successfully [[18](#_bookmark46)].

53 However, the depth of the MLP model is limited by the vanishing gradient problem,

54 making it difficult to train deeper networks. To solve the problem of vanishing gradient,

55 in 1991, Sepp Hochreiter introduced skip connections or residual connections in the

56 long short-term memory (LSTM) recurrent neural network to solve this problem [[29](#_bookmark57)].

57 Subsequently, in 2015, the concept of Highway Networks was proposed, applying the

58 concept of forget gates in LSTM to the feedforward neural network, allowing information

59 to spread in the network and alleviating the vanishing gradient problem [[30](#_bookmark58)]. Then,

60 based on Highway Networks, ResNet further simplifies the structure, removes forget

61 gates, and uses simple skip connections directly, so that signals can be propagated

62 directly without the intervention of the gating mechanism. This structure has proven

63 to be very effective in training very deep neural networks [[31](#_bookmark59)].

64 This work aims to enhance the applicability of the mlp model so that it can adapt

65 to more situations. The combination of the MLP and ResNet can synergizing the

66 strengths of two models, making the model has greater versatility and stability.

67 In the rest of the paper, the theory of MLP-ResNet and its solution will be shown

68 in Section [2](#_bookmark0); applications in 3 real-world cases in energy field will be represented in

69 Section [3](#_bookmark17); the conclusion of this paper is shown in Secton [4](#_bookmark28).

# 70 2 Theoretical Framework

## 71 2.1 Knowledge Background

### 72 2.1.1 Multiple Layer Perceptron (MLP)

73 Formally, an MLP consists of an input layer, one or more hidden layers, and an

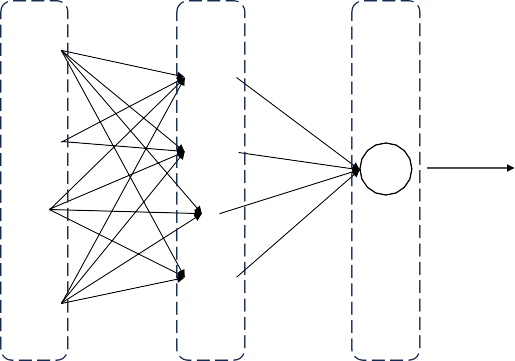
74 output layer. Each layer is composed of numerous artificial neurons, also referred

75 to as perceptrons or nodes, interconnected via weighted connections. The primary

76 function of the MLP is to transform input data through successive layers of nonlinear

77 transformations, ultimately producing an output prediction. A simple mlp network

78 structure with one-hidden-layer is shown in Fig.([1](#_bookmark1)).



Feature 1

Feature 2

Feature n

output

Input Layer

Hidden Layer

Output Layer

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

79 Mathematically, the forward propagation process of an MLP can be expressed as

80 follows: for each layer *l*, the output x(*l*) is computed as the application of a nonlinear

81 activation function *σ* to the linear transformation of the previous layer’s output x(*l*−1),

82 incorporating weights W(*l*) and biases b(*l*) :

z(*l*) = W(*l*)x(*l*−1) + b(*l*)

x(*l*) = *σ* z(*l*)

(1)

83 During training, the parameters (weights and biases) of the MLP are optimized to

84 minimize a predefined loss function, typically through backpropagation and gradient-

85 based optimization techniques. Backpropagation involves the systematic calculation of

86 gradients with respect to the parameters of the network, facilitating parameter updates

87 in the direction that reduces the loss.

88 MLPs are characterized by their universal approximation capabilities, enabling

89 them to approximate arbitrary functions with sufficient capacity and data. However,

90 their effectiveness is contingent upon various factors, including network architecture

91 design, activation functions, optimization algorithms, and hyperparameter tuning.

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

### 2.1.2 Residual Network (ResNet)

Recent advancements in the field of image recognition have underscored the crit- ical role of network depth, particularly in Convolutional Neural Networks (CNNs), as elucidated by recent studies [[32](#_bookmark60)]. 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 pioneer- ing solution by introducing a residual learning framework. Unlike conventional CNNs where each layer aims to directly learn the underlying target function *H*(*x*), 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 informa- tion of *x* in *H*(*x*). Distinguishing between the architectural setups of conventional CNN blocks and ResNet blocks is shown in Fig.([2](#_bookmark2)). 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.

Weight Layer

Weight Layer



Weight Layer

Activation Function

Weight Layer

**+**

Activation Function



Activation Function

Activation Function



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

110 The core architectural feature of ResNet is the incorporation of "shortcut connec-

111 tions" or "skip connections," which facilitate identity mapping. Through these connec-

112 tions, the original input *x* is added directly to the output of the stacked layers, thereby

113 enabling the flow of information without significant alteration. This mechanism not only

114 fosters smoother gradient flow during backpropagation but also mitigates the vanishing

115 gradient problem commonly encountered in deep networks.

## 116 2.2 The proposed MLP-ResNet model

### 117 2.2.1 The representation of MLP-ResNet model and its solution

118 As we mentioned before, it’s apparent that the most existing machine learning

119 models including MLP often face degradation phenomenon which means the accuracy

120

121

122

123

124

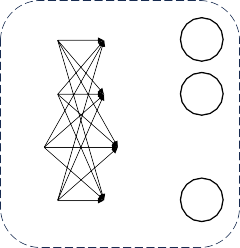
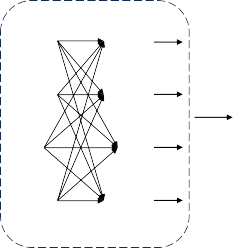
n

n

n

n

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](#_bookmark3)) and the *⃝* with number means the neuron index in each layer. The number of MLPRS BLOCK depends on the depth of MLPRS.



1

1

1

2

Feature

1

1

2

2

**+**

2

Feature 2

2

MLPRS BLOCK

Feature 1

2

MLPRS BLOCK

1

1

output

**Fig. 3.** Structure of MLPRS

n

n

n

125

126

127

128

Suppose (*X, y*) is the input data of the model which *X* has *n*1 features, it is composed of (*Xt, yt*)(*t* = 1*,* 2*, · · · , 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)*,* (2)

129 where *h*0 is the output of ResNet’s input layer.

130 When it come to the first MLPRS BLOCK, we could write the mathematical

131 expression as follows:

*h*1 = *W* (2)*h*0 + *b*(2)*,*

*h*2 = *δ*(*h*1)*, h*3 = *W* (3)*h*2 + *b*(3)*, h*4 = *W* (4)*h*3 + *h*0*,*

1

(3)

132

133

134

135

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. *δ*(*·*) is the activation function of the MLP model.

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

136

follows:

*h*5 = *W* (2)*h*4 + *b*(2)*,*

*h*6 = *δ*(*h*5)*, h*7 = *W* (3)*h*6 + *b*(3)*, h*8 = *W* (4)*h*7 + *h*4*,*

2

(4)

137

138

139

140

141

142

143

144

145

146

After the iteration of *k* depth, the k-th block’s output *h*4*k* could be obtained:

*h*4*k* = *W* (4)*h*4*n*−1 + *h*4*n*−4*.* (5)

*k*

Finally, we could calculate the output *y*ˆ of the ResNet:

*y*ˆ = *W* (5)*h*4*n* + *b*(5) (6)

In Eq.([2](#_bookmark4))([3](#_bookmark5))([4](#_bookmark6))([5](#_bookmark7))([6](#_bookmark8)), *W* (*i*)(*i* = 1*,* 2*,* 3*,* 4*,* 5) and *b*(*j*)(*j* = 1*,* 2*,* 3*,* 5) are the parame- ters of neural network in MLPRS. *W* (4)(*p* = 1*,* 2*, · · · , k*) is the k-th element in *W* (4).The specific shape of the parameter matrix are shown in Tab.[1](#_bookmark9).

*p*

**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 applica- tion 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 *δ*(*·*) could be express as follows:

*δ*(*x*) =

*ex − e*−*x*

*ex* + *e*−*x*

(7)

### 147 2.2.2 Adam algorithm for training the MLP-ResNet model

148 Normally, the neural network could not obtain an analytical solution, and the

149 proposed MLPRS model in this paper is no exception. So we need to use optimization

150 algorithms to get its solution such as Gradient Descent [[33](#_bookmark61)], Stochastic Gradient Descent

151 [[34](#_bookmark62)], Adam [[35](#_bookmark63)]. In this paper, we introduce the Adam algorithm to train the proposed

152 model due to its efficiency, robustness, and adaptability.

153 First, we need to define the training error *et* at each point (*Xt, yt*):

*et* = *yt −* (*W h*4*n* + *b* )*,* (8)

(5) (5)

154 Thus, we could obtain the sum of training error:

*E*(***W*** *,* ***b***) = 1 Σ *e*2 = *eT e,* (9)

*n*

*n*

*t*

*t*=1

155 where ***W*** is composed of *W* (*i*) and ***b*** is composed of *b*(*j*). ***W*** and ***b*** are the parameters

156 whcih need to be solved by Adam.

157 Then, in order to complete Adam, we need to finish the gradient descent. So we

158 have to get the gradient of *E*(***W*** *,* ***b***):

159

where ***L*** is the gradient.

*∂E*

***L*** = [ *,*

*∂****W***

*∂E* ]*,* (10)

*∂****b***

160

161

162

163

164

Different with ordinary gradient descent, it introduces the concept of the modified bias-corrected first moment estimate *m*ˆ *t* and bias-corrected second raw moment estimate *v*ˆ*t* to speed up convergence, their mathematical expressions are as follows:

*mt* = *µ*1 *· mt*−1 + (1 *− µ*1) *·* ***L****,* (11)

*vt* = *µ*2 *· vt*−1 + (1 *− µ*2) *·* ***L***2*,* (12)

= *,* (13)

165

*m*ˆ *t*

*v*ˆ

*t*

*mt*

*t*

1 *− µ*

1

*vt*

= *,* (14)

1 *− µt*

2

166 where *µ*1 and *µ*2 mean decay rates which are used to control the decay speed of the

167 first and second moments of the gradients, respectively.

168 Finally, we can obtain the iterative formula:

" ***W*** *t*+1 #

=

*— l*1 *· √v*ˆ

*.* (15)

+ *ϵ*

***b****t*+1

" ***W*** *t* #

***b****t*

*m*ˆ *t*

*t*

169

170

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](#_bookmark16):

**Algorithm 1:** The complete process of Adam training MLPRS **Input:** *E*(***W*** *,* ***b***) (Eq.([9](#_bookmark11))), Learning rate *l*1, max\_epochs **Initialize:** [***W*** *,* ***b***] *→ random*();

*µ*1 *→* 0*.*9; *µ*2 *→* 0*.*999;

*m*0 *→* 0; *v*0 *→* 0;

*epoch →* 0;

**1 while** *epoch < max\_epochs* **do**

171

**2** ephoch = ephoch + 1 ;

**3 *L*** *→* Eq.([10](#_bookmark12));

**4** *mt, vt →* Eq.([13](#_bookmark13))([14](#_bookmark14));

"^ ^

***W*** *t*+1

**5**

***b****t*+1

# *→* Eq.([15](#_bookmark15));

**6 end**

**7 return** [***W*** *,* ***b***]

### 172 2.2.3 Optimal Model Parameter Selection Using Gridsearch algorithm

173 In Section [2.2.2](#_bookmark10), we obtain the parameter set [***W*** *,* ***b***] through Adam algorithm, but

174 the depth *k*, learning rate *l*2 and the number of nunber of neurons *nr*(*r* = 2*,* 3) are still

175 need to be tuned. Here we introduce the Gridsearch algorithm to tune the parameters.

176 The basic principle of GridSearch is to exhaustively search all possible parameter

177 combinations in the parameter space, then perform cross-validation on each parameter

178 combination, and select the parameter combination with the best performance.

179 Suppose we have a model parameter space Θ, where each parameter combina-

180 tion can be represented by a vector ***θ*** which consisting of *k, l*2*, n*2*, n*3. Our goal is to

181 find the best parameter combination ***θ***∗ given the training data set *D*train, so that the

182 model can perform better on the validation data set *D*val for optimal performance. Its

183 mathematical principle can be expressed as follows:

***θ***∗ = arg min*f* (***θ****, D*train*, D*val)*,* (16)

***θ***∈Θ

184 where the term *f* (***θ****, D*train*, D*val) represents the performance metric obtained by training

185 the model with parameter combination ***θ*** on the validation set *D*val. Here we use

186 Negative mean square error (NMSE) to calculate the metric:

1 Σ 2

*f* (***θ****, D*train*, D*val) = *−* (*yi − y*ˆ*i*) (17)

*|D*val*| i*∈*D*al

187 **3 Applications**

188 To verify the applicability and stability of the proposed model, we use 3-real-world

189 data in energy field. The data set has important practical significance and it will be

190 discussed further in later context. Moreover, min-max mapping is used to prevent

191 overflow here. The MAPE metric are used to measure the performance of the model

192 and its mathematical expression is as follows:

1 Σ *|y*ˆ(*t*) *− y*(*t*)*|*

*s*

*k*∈*U*

*|y*(*t*)*|*

(18)

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

194 10 models are used to compare, and the information is shown in Tab.[2](#_bookmark18).

**Table** 2: Information of models used for comparison

|  |  |  |  |
| --- | --- | --- | --- |
| **Full Name** | **Abbreviation** | **Reference** | **Year** |
| Gated Recurrent Unit | gru | [[19](#_bookmark47)] | 2017 |
| Random Forest Regression | rf | [[20](#_bookmark48)] | 2001 |
| Extreme Gradient Boosting | xgb | [[21](#_bookmark49)] | 2015 |
| Long Short-Term Memory | lstm | [[22](#_bookmark50)] | 2000 |
| Support Vector Regression | svr | [[23](#_bookmark51)] | 1996 |
| Convolution Neural Network | cnn | [[24](#_bookmark52)] | 2015 |
| Multilayer Perceptron | mlp | [[25](#_bookmark53)] | 2009 |
| CNN-LSTM | cnnlstm | [[26](#_bookmark54)] | 2019 |
| Convolutional LSTM | convlstm | [[27](#_bookmark55)] | 2017 |
| General Regression Neural Network | grnn | [[28](#_bookmark56)] | 2004 |

## 195 3.1 Case 1:Electricity Transformer Oil Temperature

196 In power distribution problems, the accuracy of voltage distribution is crucial. The

197 distribution of electricity needs to be adjusted according to the needs of different re-

198 gions, and this adjustment often depends on the continuous use of electricity. However,

199 predicting future demand in a specific region is a difficult task as it is affected by various

200 factors such as working days, holidays, seasons, weather, temperature, etc. Any incor-

201 rect prediction may damage the operation of the electrical transformer. As a result,

202 there is currently no very effective way to predict future electricity usage, and managers

203 are forced to make decisions based on empirical numbers, which are often higher than

204 actual demand. This results in unnecessary waste and depreciation of electricity and

205 equipment.

206 The transformer oil temperature can reflect the operating status of the electrical

207 transformer, so in this article we use oil temperature prediction to better solve the

208 voltage distribution problem to avoid unnecessary waste.

209 In this paper, we collect hourly transformer oil temperature data from 1:00 on June

210 1, 2018 to 19:00 on June 26, 2018 from the website *https://github.com/zhouhaoyi/ETDataset*.

211 The first 496 points are used to train and the rest 124 points are used to test. The

212 training plot is shown in Fig.[4](#_bookmark19) and the testing plot is shown in Fig.[5](#_bookmark20). The MAPE value

213 of all the models are presented in Table [3](#_bookmark21).

214 From the training and testing plot, we could observe that all the models have

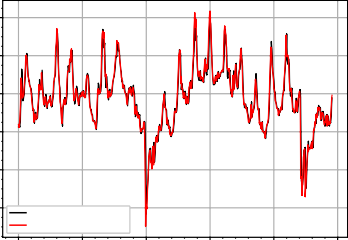
215 good performance and the prediction curve are very close to the curve of raw data. In

216 addition, from Tab.[3](#_bookmark21), it’s clearly that the proposed model has the smallest MAPE value

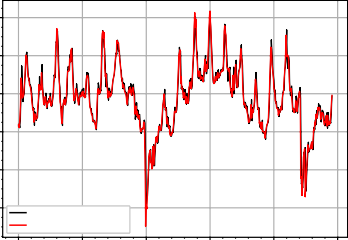
217 in testing. Only the testing values of gru and svr are close to MLP-ResNet, but both

218 of them perform worse than the proposed model in training. The rf model has the best

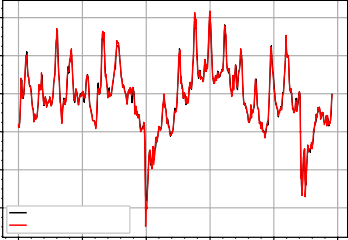
219 MAPE value in training but perform bad in testing.



!

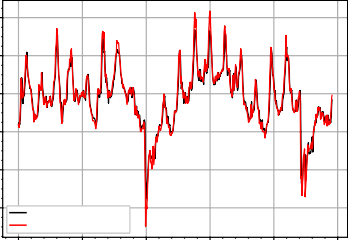


!



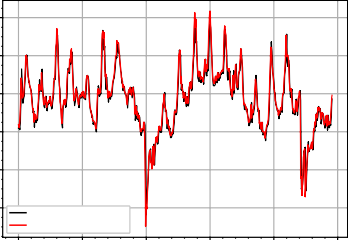
!

" !

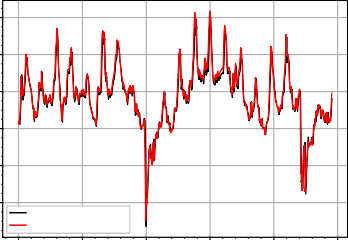


"

!

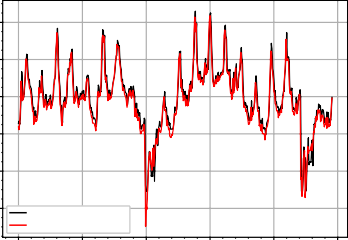


!

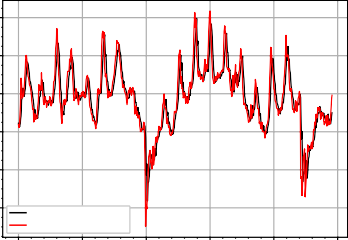


!

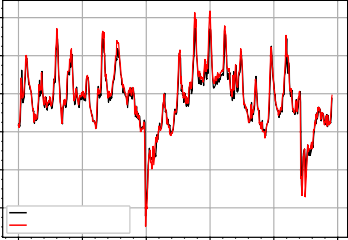
!



!

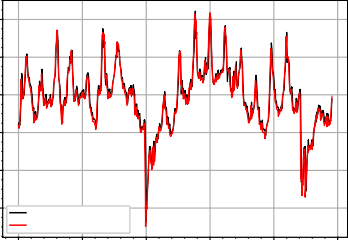


!



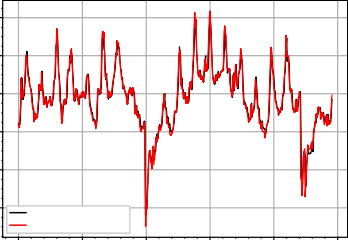
!

!



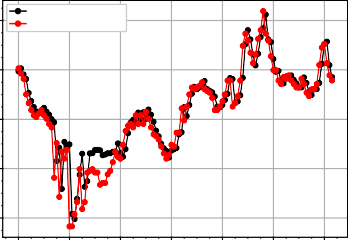
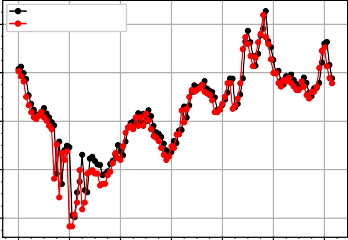
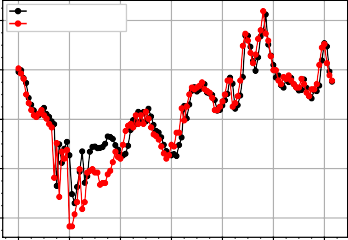
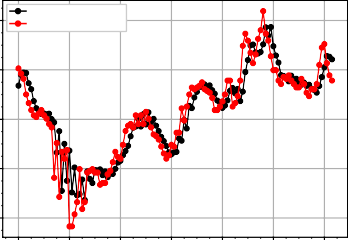
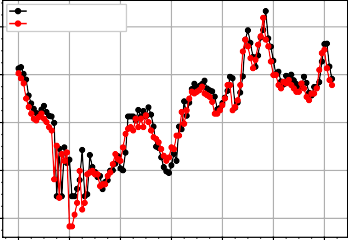
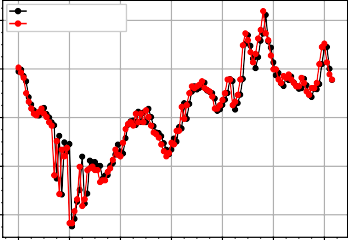
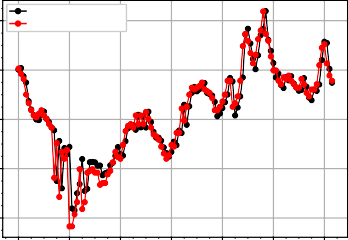
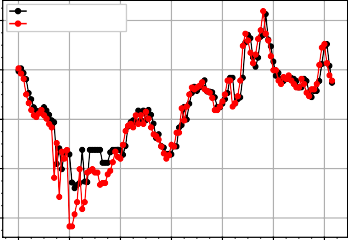
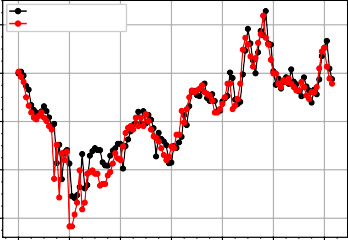
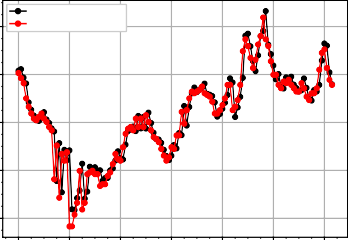
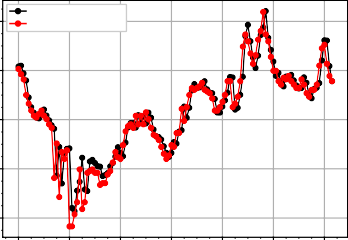
!

!



!

**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 |

## 220 3.2 Case 2:U.S. Imports of Crude Oil and Petroleum Products

221 Petroleum and its products are a key component of the global economy and have

222 profound impacts on energy markets, trade balances and geopolitics. Therefore, accu-

223 rate forecasts and analysis of U.S. crude oil and petroleum product imports are critical

224 to the stability of international energy markets and the development of the global econ-

225 omy.

226 In this paper, we collect the monthly data from January 15, 1981 to June 15, 2023,

227 and the data comes from the U.S. Energy Information Administration. For convenience

228 of expression, we called the data as ICOP. The first 408 points are used for training

229 and the last 102 points are used for testing. The training and testing plot are shown in

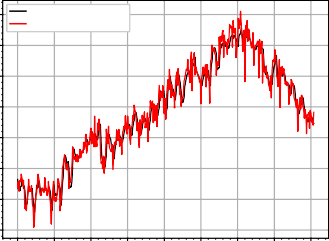
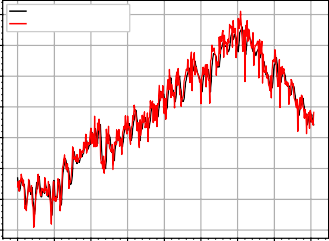
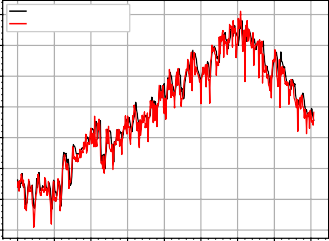
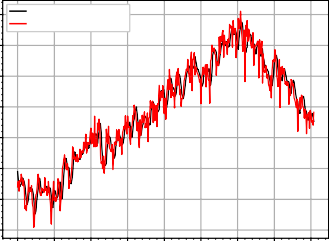
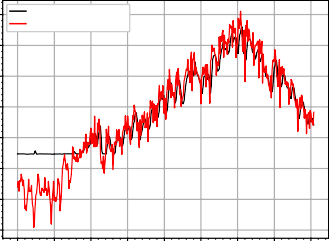
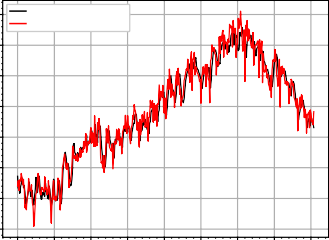
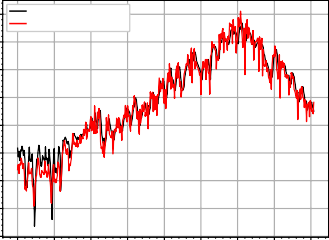
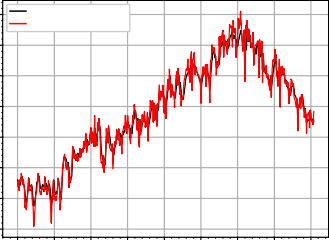
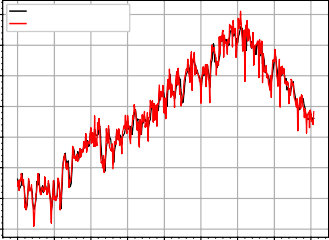
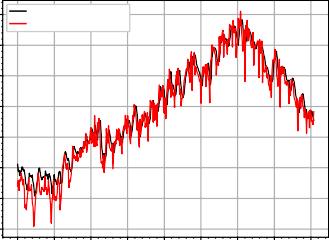
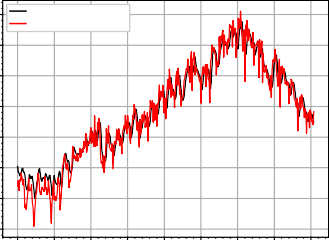
230 Fig.[6](#_bookmark22) and Fig.[7](#_bookmark23). The MAPE value are shown in Tab.[4](#_bookmark24).

231 All models’ curves shown in Fig.[6](#_bookmark22) and Fig.[7](#_bookmark23) also fit well both in training and

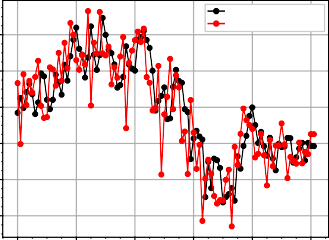
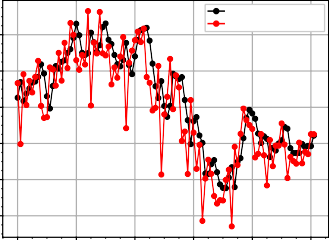
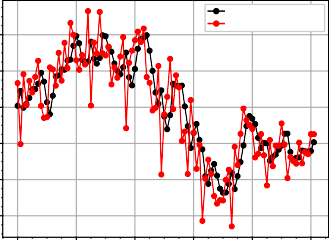
232 testing. From the MAPE table, while it is evident that MLP-ResNet performs best

233 during testing, its performance during training is suboptimal. The rf model achieves

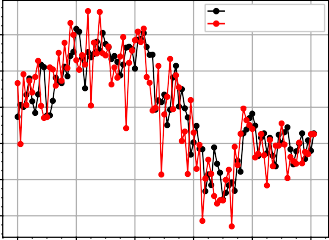
234 the best MAPE value during training but performs poorly in testing.



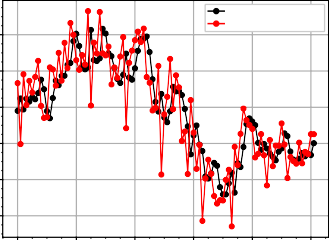
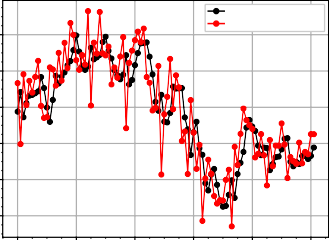
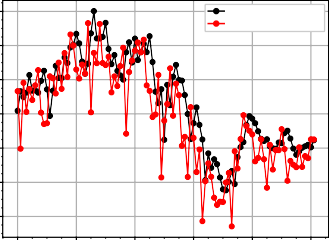
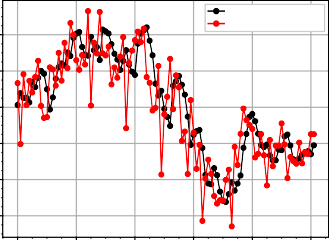
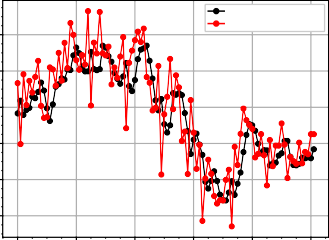
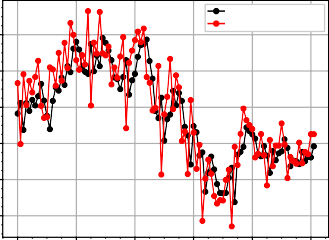
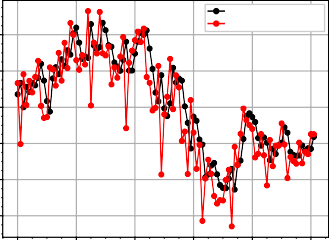
**Fig. 6.** Prediction values of ICOP training set



!



!



**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 |

## 235 3.3 Case 3:Inland Wind Turbine Power Generation

236 Wind energy is of great significance to combat climate change, reduce carbon emis-

237 sions, and achieve sustainable energy development. Forecasts of inland wind turbine

238 power generation can unveil the potential and feasibility of wind power generation in

239 the region. Similarly, comprehending the power generation of individual turbines in

240 inland wind farms aids in optimizing energy production and supply. By predicting

241 wind power generation and adjusting turbine operating parameters, energy utilization

242 efficiency can be enhanced, power generation costs can be minimized, and sustainable

243 energy production can be achieved.

244 Here, we use this hourly Inland Wind Turbine Power Generation data (IWTPG)

245 which is from September 11 to October 7, 2015, and the data is found from the website

246 *https://zenodo.org/records/5516539*. We use the first 420 points to train and the last

247 105 points to test. The training and testing plot are shown in Fig.[8](#_bookmark25) and Fig.[9](#_bookmark26). The

248 performance of the models are shown in Tab.[5](#_bookmark27).

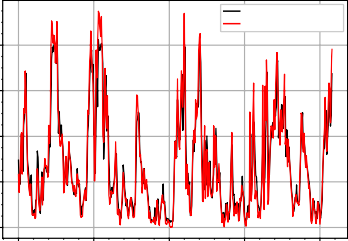
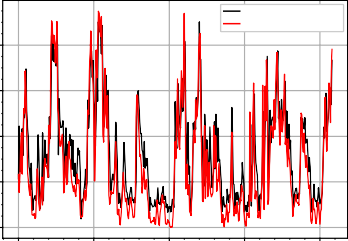
249 As the MAPE values shown in Tab.[5](#_bookmark27), it’s apparent that MLP-ResNet has a sig-

250 nificantly lower training MAPE than the other models. Although MLP achieves the

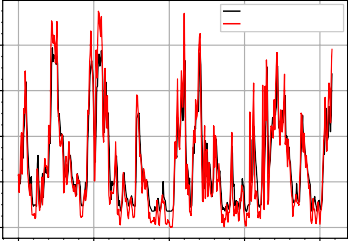
251 second-best testing MAPE, its performance in training is much worse than that of MLP-

252 ResNet. Similar to the other two cases, the rf model also achieves the best training

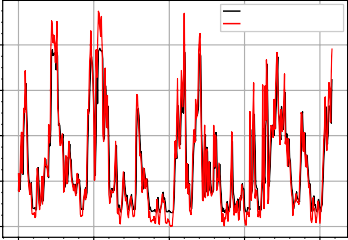
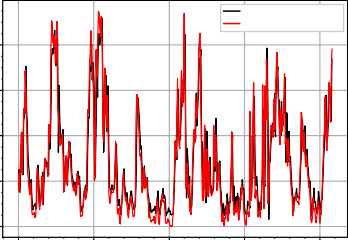
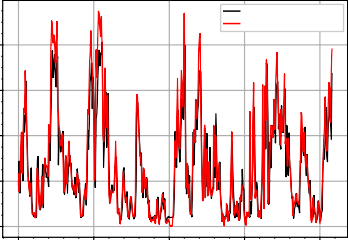
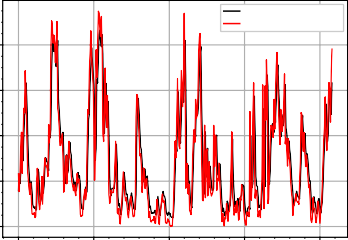
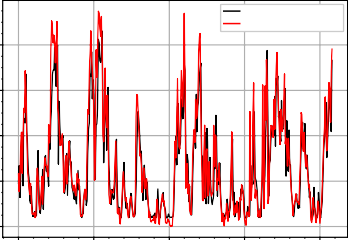
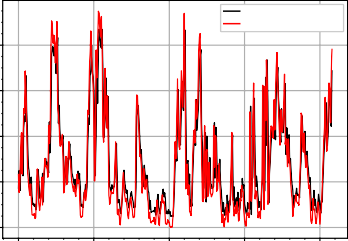
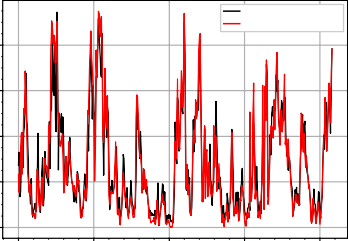
253 MAPE value but performs poorly in testing.



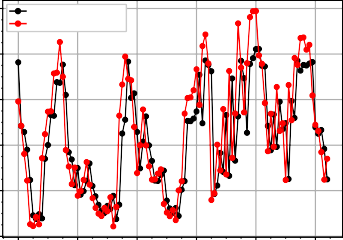
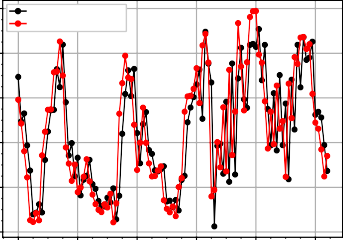
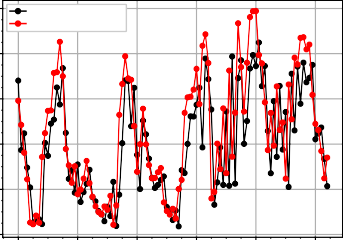
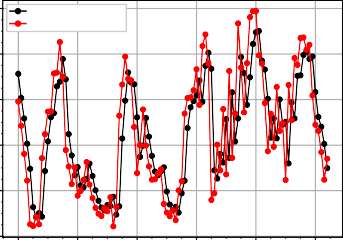
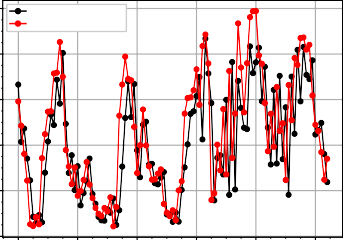
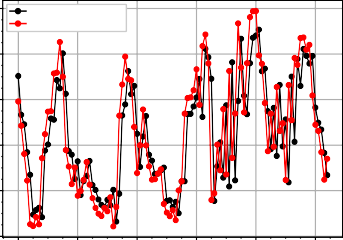
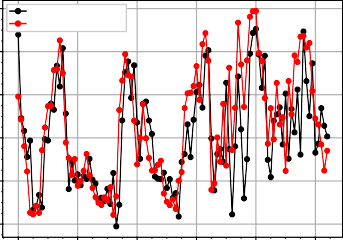
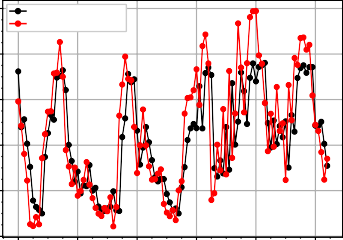
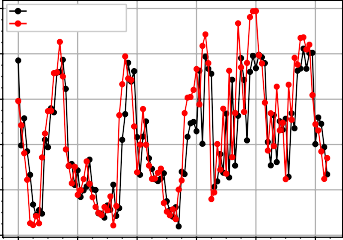
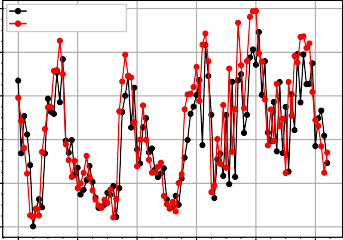
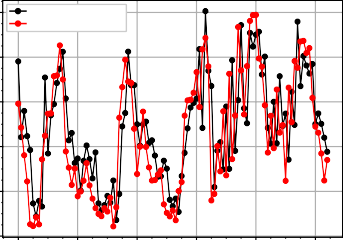
!



!



**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 |

254 **3.4 Disscusion**

255 Clearly, the proposed MLP-ResNet model performance on the training set is not

256 outstanding, but it performs the best in testing in 3 cases and the prediction curve of

257 MLP-ResNet is very close to the raw curve. This shows that it has strong generalization

258 ability and prediction accuracy. This result shows the reliability and stability of the

259 model in real scenarios. Meanwhile, the rf model achieves the best training MAPE

260 value, but it frequently performs worse in testing. The reason for this may be that the

261 model is not complex enough or overfitting occurs.

262 **4 Conclusions**

263 In this paper, we introduce the MLP-ResNet model, presenting a comprehensive

264 theoretical framework, model training methodology, and parameter tuning approach.

265 Moreover, according to 3 cases in Sec.[3](#_bookmark17), it shows that the proposed model often performs

266 best in testing and has good versatility. The way combines the ResNet and MLP could

267 effectively improve the prediction accuracy and applicability. We believe that this kind

268 of approach could have deeper research in the future.

269 **References**

270 [1] Rosenblatt F. The perceptron: a probabilistic model for information storage and

271 organization in the brain[J]. Psychological review, 1958, 65(6): 386.

272 [2] Rosenblatt F. Principles of neurodynamics: Perceptrons and the theory of brain

273 mechanisms[M]. Washington, DC: Spartan books, 1962.

274 [3] Schmidhuber J. Annotated history of modern AI and Deep learning[J]. arXiv

275 preprint arXiv:2212.11279, 2022.

276 [4] Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: theory and applica-

277 tions[J]. Neurocomputing, 2006, 70(1-3): 489-501.

278 [5] Ivakhnenko A G, Lapa V G. Cybernetic predicting devices[J]. (No Title), 1965.

279 [6] Ivakhnenko A G, Lapa V G. Cybernetics and forecasting techniques[J]. (No Title),

280 1967.

281 [7] Amari S. A theory of adaptive pattern classifiers[J]. IEEE Transactions on Elec-

282 tronic Computers, 1967 (3): 299-307.

283 [8] Rodríguez O H, Lopez Fernandez J M. A semiotic reflection on the didactics of the

284 chain rule[J]. The Mathematics Enthusiast, 2010, 7(2): 321-332.

285 [9] Werbos P J. Applications of advances in nonlinear sensitivity analysis[C]//System

286 Modeling and Optimization: Proceedings of the 10th IFIP Conference New York

287 City, USA, August 31–September 4, 1981. Berlin, Heidelberg: Springer Berlin

288 Heidelberg, 2005: 762-770.

289 [10] Rumelhart D E, Hinton G E, Williams R J. Learning internal representations by

290 error propagation[J]. 1985.

291 [11] Maqbool J, Aggarwal P, Kaur R, et al. Stock Prediction by Integrating Sentiment

292 Scores of Financial News and MLP-Regressor: A Machine Learning Approach[J].

293 Procedia Computer Science, 2023, 218: 1067-1078.

294 [12] Kang Y, Xu Y, Wang X, et al. HN-PPISP: a hybrid network based on MLP-Mixer

295 for protein–protein interaction site prediction[J]. Briefings in Bioinformatics, 2023,

296 24(1): bbac480.

297 [13] Li Y, Yin Z, Zheng Y, et al. Pose Estimation of Point Sets Using Residual MLP

298 in Intelligent Transportation Infrastructure[J]. IEEE Transactions on Intelligent

299 Transportation Systems, 2023.

300 [14] Samii A, Karami H, Ghazvinian H, et al. Comparison of DEEP-LSTM and MLP

301 Models in Estimation of Evaporation Pan for Arid Regions[J]. Journal of Soft

302 Computing in Civil Engineering, 2023, 7(2).

|  |  |
| --- | --- |
| 303 [15] | Gao S, Yang W, Xu M, et al. U-MLP: MLP-based ultralight refinement network |
| 304 | for medical image segmentation[J]. Computers in Biology and Medicine, 2023, 165: |
| 305 | 107460. |
| 306 [16] | ZHAFIRA F. OPTIMIZING LONG SHORT-TERM MEMORY NETWORKS |
| 307 | WITH MULTILAYER PERCEPTRON FOR TIME SERIES PREDICTION[D]. |
| 308 | Universitas Gadjah Mada, 2023. |
| 309 [17] | Chen S A, Li C L, Yoder N, et al. Tsmixer: An all-mlp architecture for time series |
| 310 | forecasting[J]. arXiv preprint arXiv:2303.06053, 2023. |
| 311 [18] | Ghimire S, Nguyen-Huy T, Prasad R, et al. Hybrid convolutional neural network- |
| 312 | multilayer perceptron model for solar radiation prediction[J]. Cognitive Computa- |
| 313 | tion, 2023, 15(2): 645-671. |
| 314 [19] | Dey R, Salem F M. Gate-variants of gated recurrent unit (GRU) neural net- |
| 315 | works[C]//2017 IEEE 60th international midwest symposium on circuits and sys- |
| 316 | tems (MWSCAS). IEEE, 2017: 1597-1600. |
| 317 [20] | Breiman L. Random forests[J]. Machine learning, 2001, 45: 5-32. |
| 318 [21] | Chen T, He T, Benesty M, et al. Xgboost: extreme gradient boosting[J]. R package |
| 319 | version 0.4-2, 2015, 1(4): 1-4. |
| 320 [22] | Gers F A, Schmidhuber J, Cummins F. Learning to forget: Continual prediction |
| 321 | with LSTM[J]. Neural computation, 2000, 12(10): 2451-2471. |
| 322 [23] | Drucker H, Burges C J, Kaufman L, et al. Support vector regression machines[J]. |
| 323 | Advances in neural information processing systems, 1996, 9. |
| 324 [24] | O’Shea K, Nash R. An introduction to convolutional neural networks[J]. arXiv |
| 325 | preprint arXiv:1511.08458, 2015. |
| 326 [25] | Popescu M C, Balas V E, Perescu-Popescu L, et al. Multilayer perceptron and |
| 327 | neural networks[J]. WSEAS Transactions on Circuits and Systems, 2009, 8(7): |
| 328 | 579-588. |
| 329 [26] | Kim T Y, Cho S B. Predicting residential energy consumption using CNN-LSTM |
| 330 | neural networks[J]. Energy, 2019, 182: 72-81. |
| 331 [27] | Kim S, Hong S, Joh M, et al. Deeprain: Convlstm network for precipitation pre- |
| 332 | diction using multichannel radar data[J]. arXiv preprint arXiv:1711.02316, 2017. |
| 333 [28] | Kulkarni S G, Chaudhary A K, Nandi S, et al. Modeling and monitoring of batch |
| 334 | processes using principal component analysis (PCA) assisted generalized regression |
| 335 | neural networks (GRNN)[J]. Biochemical Engineering Journal, 2004, 18(3): 193- |
| 336 | 210. |
| 337 [29] | Hochreiter S. Untersuchungen zu dynamischen neuronalen Netzen[J]. Diploma, |
| 338 | Technische Universität München, 1991, 91(1): 31. |

339 [30] Srivastava R K, Greff K, Schmidhuber J. Highway networks[J]. arXiv preprint

340 arXiv:1505.00387, 2015.

341 [31] He K, Zhang X, Ren S, et al. Identity mappings in deep residual net-

342 works[C]//Computer Vision–ECCV 2016: 14th European Conference, Amster-

343 dam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. Springer

344 International Publishing, 2016: 630-645.

345 [32] He K, Zhang X, Ren S, et al. Deep residual learning for image recogni-

346 tion[C]//Proceedings of the IEEE conference on computer vision and pattern recog-

347 nition. 2016: 770-778.

348 [33] Andrychowicz M, Denil M, Gomez S, et al. Learning to learn by gradient descent

349 by gradient descent[J]. Advances in neural information processing systems, 2016,

350 29.

351 [34] Amari S. Backpropagation and stochastic gradient descent method[J]. Neurocom

352 puting, 1993, 5(4-5): 185-196.

353 [35] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint

354 arXiv:1412.6980, 2014.