

Search on Chiller load prediction based on CEEMDAN-BiLSTM-Attention model for sufficient data and small sample data cases

Xu Zhang
2200467

(of Affiliation)

Software Engineering and Artificial Intelligence program
2200467@student.uwtsd.ac.uk

Abstract: Maintaining indoor thermal comfort with heating, ventilation, and air conditioning (HVAC) systems requires a significant amount of energy, with chillers accounting for over 50% of the total. Therefore, a technical path based on chiller load prediction is essential to achieve low energy operation of air conditioning systems in buildings. Many machine learning methods have been widely used for load prediction, but for new buildings, there is still a lack of data to support model training for HVAC chiller load prediction. To address the above load prediction problems, this paper proposes the CEEMDAN-BiLSTM-Attention method for improving chiller load prediction accuracy. The final experimental findings are contrasted with the traditional prediction models like CNN, LSTM and hybrid models (CNN-LSTM, CNN-BiLSTM) and so on. Meanwhile, as for the case of small data samples, this paper builds two migration models CNN-LSTM-TL and CNN-BiLSTM-TL for the comparison. The results demonstrate that the CEEMDAN-BiLSTM-Attention model outperforms the other models in predicting chiller load for both buildings with sufficient data and buildings with small sample data.

Keywords: HVAC system, Chiller load forecast, CEEMDAN decomposition, BiLSTM neural network, Attention mechanism, Machine learning, Migration model.

1. INTRODUCTION

In recent years, there has been an expanding concern about energy consumption, with energy consumption for buildings reaching about 40% [1]. In order to achieve intelligent decision management in terms of energy, the ability to correctly forecast a building's energy load is essential [2].

For example, in the case of office buildings, the percentage of energy lost in heating and cooling operation can be up to 68.01% [3]. In HVAC systems, chillers consume more than half of the energy required to maintain indoor thermal comfort, and the majority of instantaneous load-based supervisory optimization solutions have a lag [4]. Therefore, it is necessary to develop HVAC system power distribution strategies based on chiller load forecasts to achieve energy efficiency optimization.

In the past decades, many load forecasting models have been studied, such as exponential smoothing model

was used for computer equipment load prediction to achieve computing resource allocation and improve computer system revenue [5]. Artificial Neural Network (ANN) was used for two different load characteristics of weekdays and weekends to forecast the electricity load separately and take the results as a basis for electricity distribution [6]. Moreover, the input data even can be converted into an image-based format and used a Convolutional Neural Network (CNN) prediction model for load prediction to improve the prediction results [7][8]. Additionally, a strategy for projecting the short-term integrated energy system demand based on time series features was put out in the literature [9]. However, using the above forecasting models, there may be problems such as the vast amount of load data from the past, the randomness of load, and the low prediction accuracy of a single forecasting model. To address these problems, on the basis of a hybrid model CNN and Long Short-term Memory (LSTM), a new technique for predicting short-term load has been developed. The CNN-LSTM hybrid model can reduce the overall model complexity and computational effort. It has found that Empirical Modal Decomposition (EMD) is effective for nonlinear and nonstationary signals, such as time series for energy usage [11]. The combination of EMD with traditional time series analysis models can be effectively utilized to increase forecast accuracy. The XGboost algorithm was first applied for feature importance assessment in [12], and then it constructed an EMD-LSTM short-term load prediction model based on the advantages of EMD decomposition mentioned above. Although this method is effective when applied to load prediction, EMD still experiences mode aliasing brought on by intermittent signals [13]. The Ensemble EMD (EEMD) approach adds white noise of finite amplitude to each Intrinsic Mode Function (IMF) component of the decomposition to solve the mode aliasing problem [14]. However, white noise to a certain extent is always present in the IMF component, which may lead to reconstruction errors and affect the subsequent analysis and processing of the signal [15]. Although by increasing the number of white noise averages, the noise can be removed, this operation leads to an increase in computation time [16].

To enhance the effectiveness of EEMD, the Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) technique was proposed [17]. The CEEMDAN approach applies adaptive noise to each stage of the EMD decomposition, greatly improving the results and accelerating computing.

A good result can be obtained for predicting building energy consumption using a hybrid RF-LSTM model based on modal decomposition of the original data by CEEMDAN [18].

Meanwhile, for new buildings, traditional machine learning models are difficult in achieving good predictions due to limited data [19]. To solve this problem, on the one hand, LSTM and Domain Adversarial Neural Network (DANN) were combined into a hybrid model and applied to buildings with a lack of samples [20]. On the other hand, a migration model CNN-LSTM was mentioned, it can be used to solve the lack of sample problem. All of the aforementioned techniques need a lot of raw datasets from other related buildings for model training, which may lead to poor prediction of migration models when the correlation between the raw datasets and the target datasets is not strong or when there is a lack of raw datasets. Therefore, it is also important to find a prediction method that is directly applicable to small sample datasets.

Based on the above research, this paper proposes a hybrid prediction model of CEEMDAN, Bi-directional Long and Short-Term Memory neural networks (BiLSTM) and Attention mechanism to predict chiller load for HVAC systems in adequate data buildings. And contrasts the outcomes of the forecast with conventional prediction techniques such as CNN, LSTM, BiLSTM and the hybrid models (CNN-LSTM, CNN-BiLSTM, CEEMDAN-LSTM and CEEMDAN-BiLSTM). It is also applied to chiller load prediction for new buildings with small data samples, and two migration models CNN-LSTM-TL and CNN-BiLSTM-TL are added for comparison. Finally, it shows that the CEEMDAN-BiLSTM-Attention method proposed in this paper has better results than other models in both cases of sufficient data samples as well as small data samples.

2. RELATED METHODS

2.1. CNN

If a fully connected neural network is used to process large-size images has three distinct disadvantages: (1) Firstly, spatial information is lost by expanding the image into vectors; (2) Secondly, too many parameters are inefficient and difficult to train; (3) At the same time, the large number of parameters can quickly lead to network overfitting.

The use of convolutional neural networks can solve the above three problems very well. Unlike conventional neural networks, the neurons in each layer of a convolutional neural network are arranged in 3 dimensions: width, height and depth. The width and height are well understood because the convolution itself is a two-dimensional template, but the depth in a convolutional neural network refers to the third dimension of the activation data body, not the depth of the entire network, which refers to the number of layers of the network. As an example, to understand what width, height and depth are, if the image in CIFAR-10 is used as

the input to a convolutional neural network, the dimension of that input data body is $32 \times 32 \times 3$ (width, height and depth). We will see that the neurons in the layer will only be connected to a small area in the previous layer, instead of taking a fully connected approach.

CNN is a widely used neural network in deep learning and its structure is shown in Fig. 1. It contains: input layer, convolutional layer, pooling layer, incentive layer, fully connected layer, and output layer [22]. The main learning process of this neural network is to extract the relational parameters stored in its filter weights based on the connections that exist between the inputs and outputs of the data [23]. The convolutional layer and pooling layer are two important parts of CNN, the convolutional layer is used to extract data features and the pooling layer downscales the data to further extract information from the data and reduce the computational effort [24][25]. In this paper, we mainly use convolutional layer and pooling layer for data feature extraction to build migration models. The incentive layer generally uses the RELU activation function [26], which serves to transmit the outcomes of the data processing performed by the pooling layer to the fully connected layer, then to the output layer.

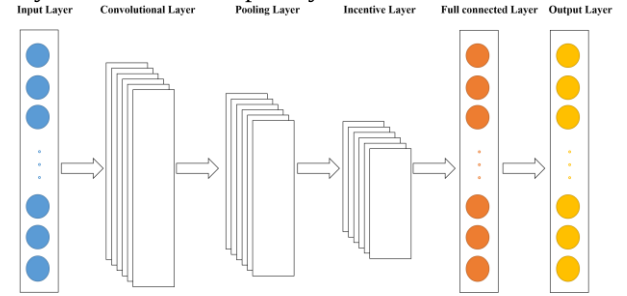


Fig. 1: CNN network structure

2.2. CEEMDAN

To address the problem of modal aliasing in the decomposed signal of EMD algorithm, EEMD and CEEMD decomposition algorithms mitigate the modal aliasing of EMD decomposition by adding pairs of positive and negative Gaussian white noise to the signal to be decomposed. However, these two algorithms always have some residual white noise in the eigenmodes of the decomposed signal, which affects the subsequent analysis and processing of the signal.

The CEEMDAN decomposition addresses the mode aliasing issue in EMD and the residual white noise problem in EEMD in two ways: (1) It adds the auxiliary noise to the decomposed IMF component rather than just adding the Gaussian white noise signal to the original signal. (2) The CEEMDAN decomposition method is to calculate the overall average of the first-order IMF components after the EMD decomposition to obtain the final first-order IMF components, and then the residual part of this component is operated as above and repeated several times until the decomposed IMF components tend to be smooth, which successfully resolves the issue of white noise transfer from high to low frequencies. Fig.

2 displays the flow chart for the CEEMDAN decomposition [17].

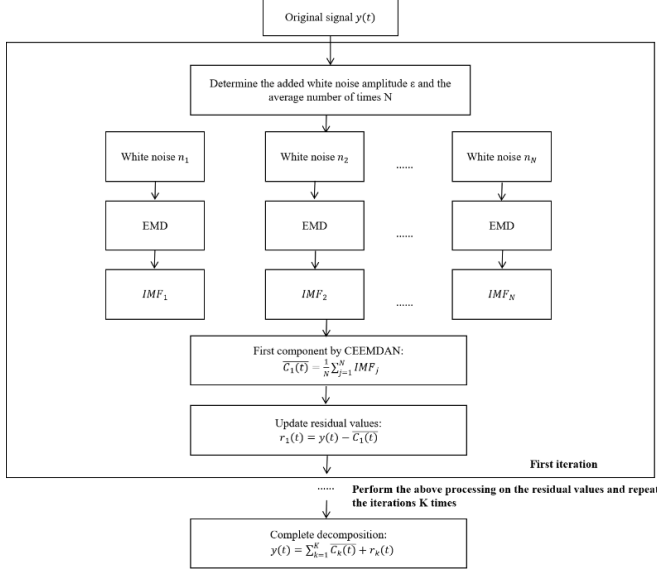


Fig. 2: CEEMDAN decomposition process

Assuming that $E_i(\cdot)$ is the i -th IMF component discovered using EMD decomposition, $\bar{C}_i(t)$ is the i -th IMF component discovered using CEEMDAN decomposition. V^j is the signal of Gaussian white noise that conforms to the normal distribution, $j = 1, 2, 3 \dots N$ is the number of times white noise is added. $y(t)$ is the signal to be decomposed, and ε is the standard table of white noise. The steps for decomposing CEEMDAN are as follows [17][27].

- i. The Gaussian white noise is added to the signal $y(t)$ to be decomposed to obtain the new signal $y(t) + (-1)^q \varepsilon v^j(t)$, where $q=1, 2$. The EMD decomposition of the new signal is performed by Eq. (1), and the first order IMF component C_1 is obtained.

$$E(y(t) + (-1)^q \varepsilon v^j(t)) = C_1^j(t) + r^j \quad (1)$$

- ii. The first IMF component of the CEEMDAN decomposition $\bar{C}_1(t)$ is obtained by averaging the N modal components obtained from the EMD decomposition, as shown in Eq. (2).

$$\bar{C}_1(t) = \frac{1}{N} \sum_{j=1}^N C_1^j(t) \quad (2)$$

- iii. The signal to be decomposed is subtracted from the first IMF component $\bar{C}_1(t)$ obtained from the CEEMDAN decomposition to get its residual $r_1(t)$, as shown in Eq. (3).

$$r_1(t) = y(t) - \bar{C}_1(t) \quad (3)$$

- iv. Eq. (4) shows how to calculate the second IMF component $\bar{C}_2(t)$ of the CEEMDAN decomposition.

$$\bar{C}_2(t) = \frac{1}{N} \sum_{j=1}^N D_1^j(t) \quad (4)$$

- v. Eq. (5) determines the residual $r_2(t)$ of the new signal less the second modal component.

$$r_2(t) = r_1(t) - \bar{C}_2(t) \quad (5)$$

- vi. Up until the resultant residual signal is a monotonic function and cannot be further decomposed, the aforementioned steps are repeated. At that point, the procedure is complete. The number of IMF components obtained at this point is K . The original signal $y(t)$ is decomposed as shown in Eq. (6).

$$y(t) = \sum_{k=1}^K \bar{C}_k(t) + r_k(t) \quad (6)$$

2.3. BiLSTM

(1) LSTM

LSTM is a special kind of recurrent neural network. Unlike feedforward neural networks, LSTMs can use time series to analyze the input; in short, when using feedforward neural networks, the neural network considers the content of the input at moment T is completely unrelated to the content of the input at time $T+1$. For many cases, such as image classification, this is fine, but for some scenarios, such as NLP (Natural Language Processing) or when we need to analyze data such as a series of photos, it is reasonable to use T or the previous input to process the $T+N$ moment is obviously a more rational use of the input information.

Traditional recurrent neural network RNN, due to the disconnection of nerve cells in the hidden layer, resulting in the inability of individual nodes to share information. Furthermore, the network could experience gradient disappearance and explosion as a result of the influence of the weight values [28]. The LSTM neural network adds three key structures to the traditional RNN: forgetting gate, input gate and output gate. Through the use of these gate structures, the model is able to selectively forget and remember the data transmitted from the previous node, effectively solve the problem of gradient disappearance and explosion in the RNN network structure. In Fig. 3, the LSTM's neural network architecture is displayed [29].

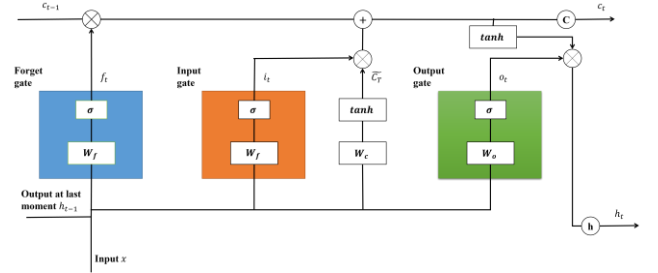


Fig. 3: LSTM neural network

The computational procedure of the LSTM neural network model is shown:

- i. The hidden state h_{t-1} of the previous moment and the input value x_t of the current moment are utilized as the input values to generate the output value f_t of the forgetting gate using Eq. (7) to calculate the output of the forgetting gate and choose the information to be forgotten.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (7)$$

- ii. Determining the input gate's output and choosing the data to be stored in memory. The output value i_t of the memory gate and the temporary cell state $\tilde{C}_t$ are obtained by Eq. (8) and (9) using the hidden state h_{t-1} of the previous instant and the input value x_t of the present moment as input quantities.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (8)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (9)$$

- iii. The parameters i_t , f_t , $\tilde{C}_t$ and the preceding moment cell state C_{t-1} are accepted as input quantities, and the current cell state C_t is calculated by Eq. (10).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (10)$$

- iv. The parameters h_{t-1} , x_t and C_t are taken as input variables in order to compute the output gate and the current moment hidden layer state. The final output gate value o_t and the state h_t of the hidden layer are then obtained by applying Eq. (11) and (12).

$$o_t = \sigma(W_o[h_{t-1}, x_t]) + b_o \quad (11)$$

$$h_t = o_t * \tanh(C_t) \quad (12)$$

(2) BiLSTM

In the one-way LSTM neural network, only the information passed from the previous node is considered, and the information passed from the next node is not taken into account. In practical model applications there is a need to collect the information of the whole input sequence, so the forward LSTM and backward LSTM are combined to form a bidirectional long and short memory network. This network can make full use of the information from the previous node and the next node to extract the backward and forward relationship of the load sequence and connect to the same output. Therefore, the model can be used to derive bidirectional temporal features from data on chiller load sequence. Input X as an input to the bidirectional LSTM layer and obtain the output after a similar calculation process as above. Fig. 4 depicts the structure [28][30].

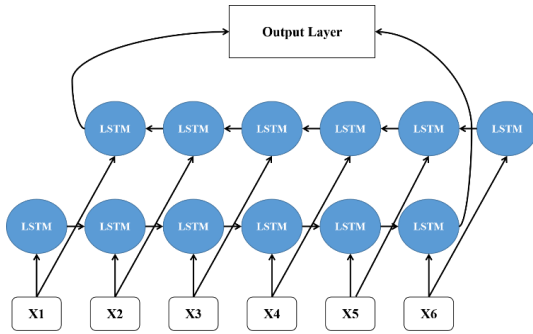


Fig. 4: BiLSTM neural network

2.4. Attention mechanism

Attention is a mechanism for improving the effectiveness of RNN (LSTM or GRU) based Encoder + Decoder models, commonly known as Attention Mechanism. For example, in machine translation and speech recognition applications, each word in a sentence is given a different weight, which makes the learning of neural network models more flexible (soft), while Attention itself can be used as an alignment relation to explain the alignment relation between translation input/output sentences and explain what exactly the model has learned, providing a window for us to open the black box of deep learning.

Attention Mechanism is very similar to the human observation mechanism of external things. When humans observe external things, they generally do not see things as a whole, but tend to selectively acquire certain important parts of the observed things according to their needs, for example, when we see a person, we tend to Attention to the person's face first, and then combine the information from different For example, when we see a person, we tend to first Attention to the

person's face, and then combine the information from different areas to form an overall impression of the observed thing. Therefore, Attention Mechanism can help the model to give different weights to each part of the input X and extract more critical and important information, so that the model can make more accurate judgments without bringing more overhead to the model's computation and storage, which is the reason why Attention Mechanism is so widely used.

The Attention Mechanism technique, which draws its inspiration from human attention, enables the model to learn and take in key informational elements. The purpose of using this method is that when a long time series is used as input, it can be weighted for features that are highly correlated with the output results. By increasing the importance of the focused information, the aim is to improve the prediction accuracy.

The total formula for the Attention mechanism is as in Eq. (13).

$$Attention(Q, K, V) = \sum_{i=1}^n \alpha_i * V_i \quad (13)$$

where Q is the Query, K is the Key, they are the feature vectors used to calculate the attention weights. V is the Value, it denotes the vector of input features. α_i is the correlation between Query and Key.

Here α_i is calculated as shown in Eq. (14) and (15).

$$\alpha_i = softmax(f(Q, K_i)) \quad (14)$$

$$f(Q, K_i) = Q^T K_i \quad (15)$$

2.5. Migration models

The core of transfer learning is to find similarities between existing knowledge and new knowledge. Since it is too expensive to learn the target domain directly from scratch, we turn to using existing relevant knowledge to assist in learning new knowledge as quickly as possible. The study of transfer learning is based on the observation that humans can apply previously learned knowledge to new problems and solve them faster or with better results. Transfer learning is given the task of learning knowledge or experience from a previous task and applying it to a new task. In other words, the purpose of transfer learning is to extract knowledge and experience from one or more source tasks and apply it to a target domain.

There are some limitations to migration learning. For example, we cannot remove the convolutional layers in a pre-trained network at will. However, due to parameter sharing, we can easily run a pre-trained network on images of different spatial dimensions. This is evident in the case of convolutional and pooling layers and because their forward function is independent of the spatial dimensions of the input content. In the case of the fully connected layer (FC), this still holds, because the fully connected layer can be transformed into a convolutional layer. So when we import a pre-trained model, the network structure needs to be the same as the pre-trained network structure, and then trained for a specific scenario and task.

In this paper, the CNN model is combined with the BiLSTM model to build the migration model, as shown

in Fig. 5. It uses superimposition of four convolutional layers and four pooling layers as CNN layers to extract data features, and then fed them into the BiLSTM neural network for chiller load prediction [31].

For the implementation of the CNN-BiLSTM-TL transfer model, at first, using a building with sufficient data to pre-train the predictive model. Then a method of freezing some of the neural network layers and retraining the unfrozen network weight parameters on a new building with small data samples is adopted to achieve the construction of the CNN-BiLSTM-TL transfer model. The process is shown in Fig. 6.

Experiments have shown that CNN-BiLSTM-TL transfer model freezes 1 CNN layer and retrain other layers of parameters to get the best prediction. The implementation of the CNN-LSTM-TL transfer model is similar to the above process, and it predicts well when the 2 CNN layers are frozen.

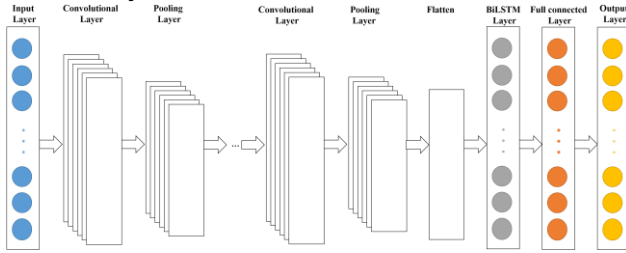


Fig. 5: CNN-BiLSTM model

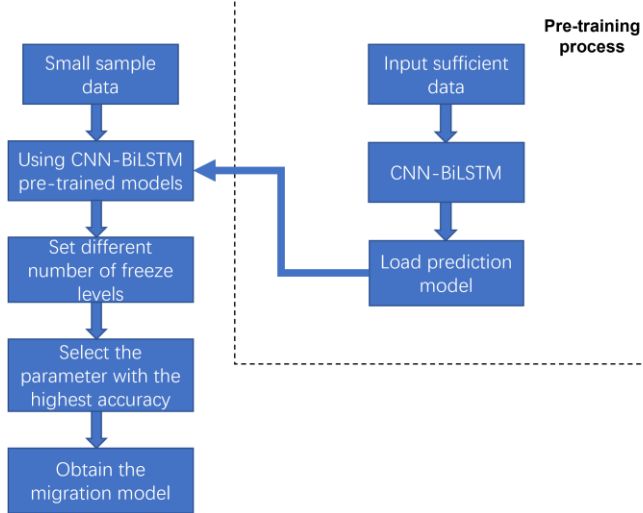


Fig. 6: CNN-BiLSTM migration model

2.6. CEEMDAN-BiLSTM-Attention

For the construction of the CEEMDAN-BiLSTM-Attention model. Firstly, the collected building chiller operation data are pre-processed by de-weighting and padding to generate original data $y(t)$, and use CEEMDAN to decompose it into multiple IMF components. After being independently normalized for each deconstructed IMF subsequence, each IMF component is then predicted using the BiLSTM and Attention method. The general flow chart of CEEMDAN-BiLSTM-Attention model is shown in Fig. 7.

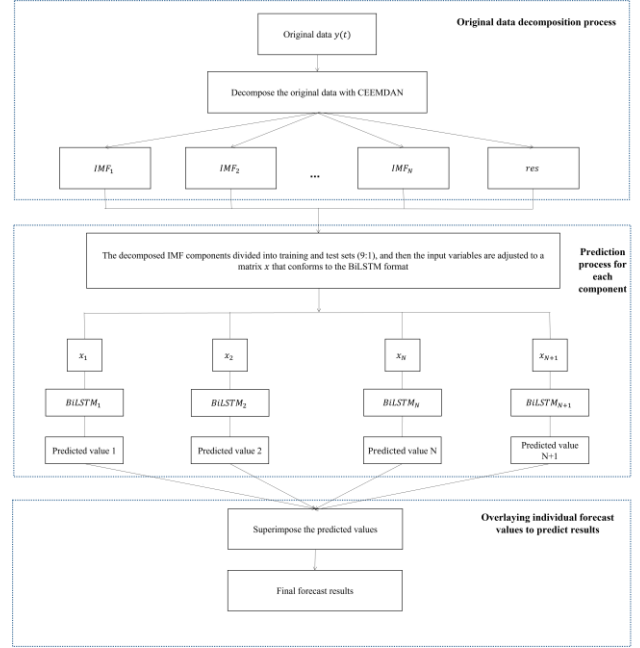


Fig. 7: Flowchart of CEEMDAN-BiLSTM-Attention

2.7. Evaluation indicators

MAPE, RMSE and MAE were selected as our evaluation indicators, which were used to measure the mean absolute percentage error, root mean square error and mean absolute error between the predicted values and actual values, respectively. These three indicators can fairly and objectively reflect the process of comparing actual data with forecast data. The calculation process of MAPE, RMSE and MAE [18] is shown in Eq. (16)(17) and Eq. (18).

$$MAPE\% = \frac{100}{n} \sum_{t=1}^n \left| \frac{y'_t - y_t}{y_t} \right| \quad (16)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y'_t - y_t)^2} \quad (17)$$

$$MAE = \frac{1}{n} \sum_{t=1}^n |y'_t - y_t| \quad (18)$$

In the above equation, y'_t is the predicted value and y_t is the actual value.

3. EXPERIMENTAL PROCESS

3.1. Description of experimental data

This paper uses data from the chiller operation of the HVAC system in buildings, which includes the load factor during chiller operation, and the formula for the load factor calculation is shown in Eq. (19), where Q_e denotes the actual cooling capacity and Q_z denotes the maximum cooling capacity (this paper uses load factor instead of load as the prediction target).

$$load\ factor = \frac{Q_e}{Q_z} \quad (19)$$

Since some parameters such as chiller cooling inlet temperature, chiller cooling outlet temperature, chiller freezing inlet temperature and chiller freezing outlet temperature have a strong correlation with the operating status of the chiller. And Load factor, as a characteristic data over time, is generally influenced by the previous moment data. Therefore, it chooses these four parameters and load factor at last hour as input variables, and let the load factor after 1 hour as output variable. Then as for the

data, it is first pre-processed to remove the outliers and duplicate values in each column, after which the average value of each column is used to fill in the missing values. In this paper, it analyzed the change in load factor for each one hour of chiller operation in these two buildings, and the change in load factor is shown in the following figures (Fig. 8 is for the sufficient data building and Fig. 9 is for the new small sample data building). From the following two figures (collected load factor time interval of one hour), it can be seen that the load factor of each building has its random and non-linear variation trend.

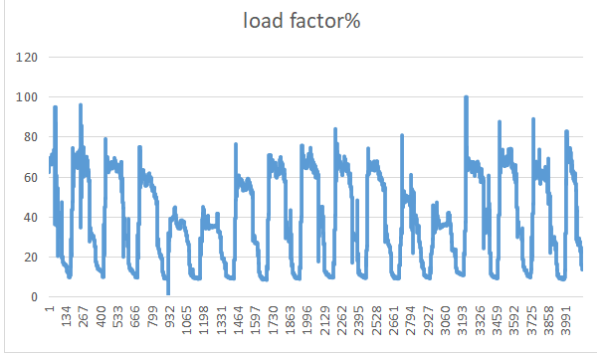


Fig. 8: Sufficient data building chiller load factor changes

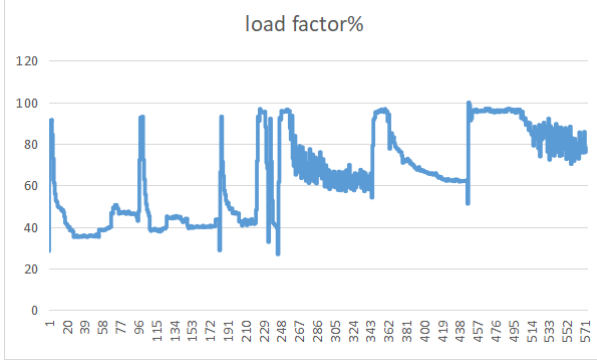


Fig. 9: Small sample data building chiller load factor changes

3.2. Experimental parameters setting

The load factor forecast after one hour is the topic of this article. As for CNN-BiLSTM migration model, after comparing the experimental results, it decides to choose the number of CNN layer $m=4$ and BiLSTM layer $n=2$. In BiLSTM structure, it has 64 hidden neurons in first layer and 32 hidden neurons in second layer. And the number of neurons in full connected layer $n_{FC}=32$. As for the CEEMDAN-BiLSTM-Attention model, it has a similar structure as BiLSTM above, and the nodes of Attention mechanism are 32. And in terms of optimizer selection, after comparing with several optimizers such as adam, radam and sgd, the results show that the adam optimizer is the best. As for the value of the batch size, 16 is used as a batch size based on the fact that the generic value of the batch size is generally powers of 2 [33]. Finally, the original data is divided into 90% as the training set and 10% as the test set. The programming environment used is python 3.6, Tensorflow 2.1, and keras 2.9.0. The model parameters are set as shown in Table. 1.

Table. 1: Parameters setting

Prediction model	Parameters setting
CNN	n_{input} : 5, m : 4, kernel size: (3*3), n_{FC} : 32, n_{batch} : 16, activation: Relu, lr: 0.0002, loss: MSE, optimizer: Adam;
LSTM	n_{input} : 5, n : 2, n_{L1} : 64, n_{L2} : 32, n_{FC} : 32, n_{batch} : 16, activation: Relu, lr: 0.0002, loss: MSE, optimizer: Adam;
CNN-LSTM	n_{input} : 5, m : 4, n : 2, kernel size: (3*3), n_{L1} : 64, n_{L2} : 32, n_{FC} : 32, n_{batch} : 16, activation: Relu, lr: 0.0002, loss: MSE, optimizer: Adam;
CNN-BiLSTM	n_{input} : 5, m : 4, n : 2, kernel size: (3*3), n_{L1} : 64, n_{L2} : 32, n_{FC} : 32, n_{batch} : 16, activation: Relu, lr: 0.0002, loss: MSE, optimizer: Adam;
CEEMDAN-BiLSTM-Attention	n_{input} : 5, m : 4, n : 2, kernel size: (3*3), n_{L1} : 64, n_{L2} : 32, n_{nodes} : 32, n_{FC} : 32, n_{batch} : 16, activation: Relu, lr: 0.0002, loss: MSE, optimizer: Adam;

4. RESULTS

4.1. Comparison of chiller load factor prediction results for buildings with sufficient data

In the first stage, the raw sufficient data are first CEEMDAN decomposed, as shown in Fig. 10, and the frequencies of the IMF components of the CEEMDAN decomposition are arranged from high to low. It can be seen that the IMF1-IMF11 curves are increasingly moving toward periodicity and regularity. The last residual component of the decomposition shows an overall trend of the data.

In the second stage, the BiLSTM neural network and Attention mechanism are used to forecast each of the deconstructed IMF components separately, and the results are stacked to create the prediction curves illustrated in Fig. 11. It is clear that the prediction curve is similar to the real curve and has less variation, demonstrating the method's excellent predictive ability.

Comparing the proposed hybrid model CEEMDAN-BiLSTM-Attention with other prediction methods. Including CNN, LSTM, two hybrid models based on Convolutional Neural Networks (CNN-LSTM) and (CNN-BiLSTM) and two hybrid models of CEEMDAN-LSTM and CEEMDAN-BiLSTM. Based on the comparison of the results in Table. 2, as can be observed, the RMSE, MAE and MAPE assessment indicators show that the CEEMDAN-BiLSTM-Attention model has the best prediction results.

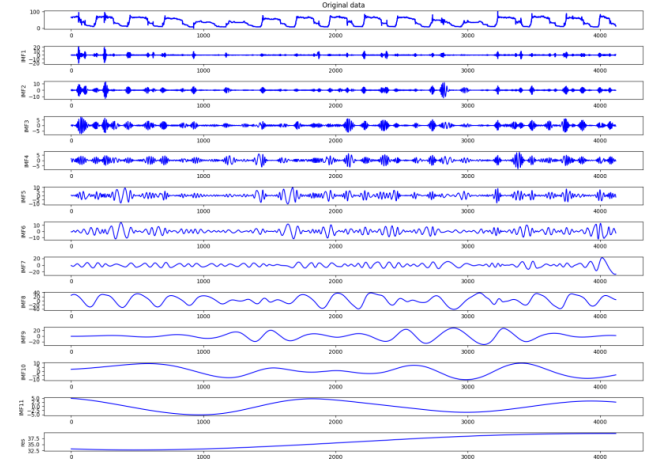


Fig. 10: CEEMDAN decomposition of sufficient data

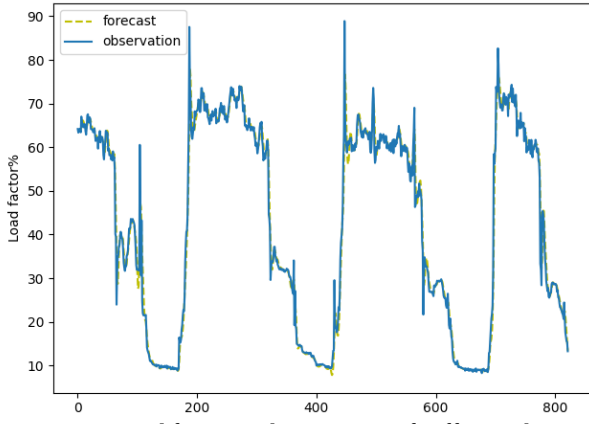


Fig. 11: Load factor prediction curve of sufficient data

Table. 2: Comparison of predictions of various models with sufficient data

models\evaluation merits	RMSE	MAE	MAPE
CNN	5.41	4.06	9.89
LSTM	4.56	3.42	9.38
CNN-LSTM	4.02	2.92	7.46
CNN-BiLSTM	3.70	2.57	6.51
CEEMADAN-LSTM	2.68	1.80	4.10
CEEMADAN-BiLSTM	2.53	1.50	3.92
CEEMADAN-BiLSTM-Attention	2.27	1.35	3.65

4.2. Comparison of chiller load factor prediction results for new buildings with small sample data

Similar to the above steps for comparing the predicted results of chiller load factor for buildings with sufficient data, the small sample data of new buildings were first CEEMDAN decomposed to produce six IMF components as shown in Fig. 12, with frequencies still arranged from high to low. As above, a general trend of the data can be observed through the residual component. After decomposing the original data, the BiLSTM neural network and Attention mechanism are then used to predict each IMF component, and each prediction is combined to create the final forecast outcome. The prediction curve of CEEMDAN-BiLSTM-Attention is shown in Fig. 13.

The difference from the previous comparison scheme is that the comparison of the results of chiller load factor prediction for new buildings with small sample data adds migration model of CNN-LSTM-TL and CNN-BiLSTM-TL. The prediction results of the two migration models are shown in Fig. 14, Fig. 15. The final comparison of the prediction results of various models is shown in Table. 3. It can be seen that the values of the three metrics of CEEMDAN-BiLSTM-Attention model are still the smallest in the case of small sample prediction. From the forecast curve, we can see that the first half of the validation set CNN-BiLSTM-TL model fits the actual results more closely. However, for the latter half of the Validation set, the CEEMDAN-BiLSTM-Attention model works better. And from the overall view

CEEMDAN-BiLSTM-Attention model can better respond to the trend of load factor change.

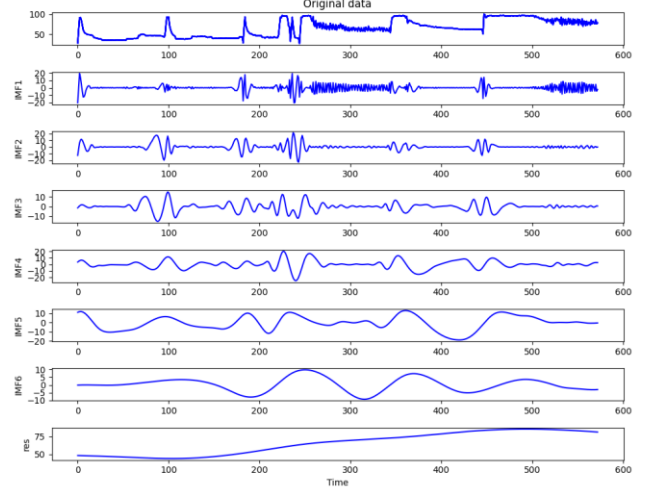


Fig. 12: CEEMDAN decomposition of small sample data

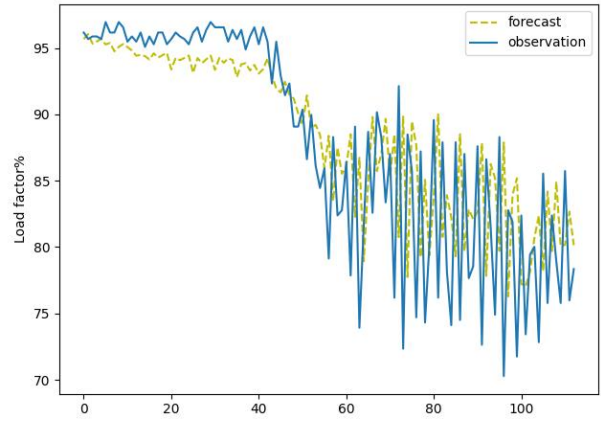


Fig. 13: CEEMDAN-BiLSTM-Attention load factor prediction curve of small sample data

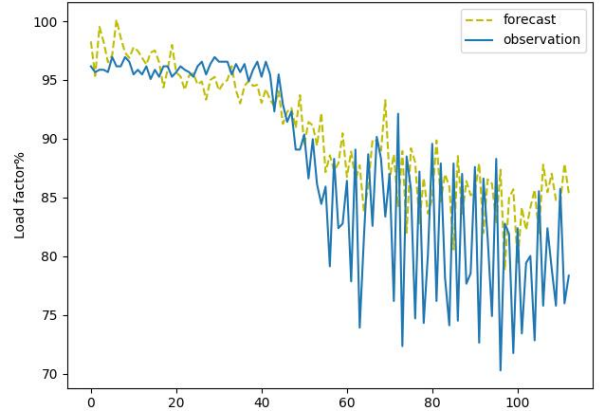


Fig. 14: CNN-LSTM-TL load factor prediction curve of small sample data

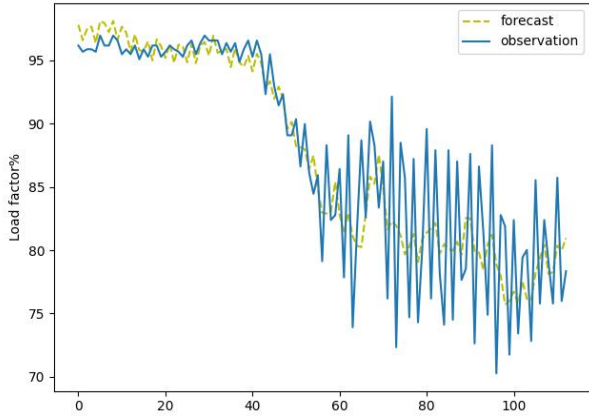


Fig. 15: CNN-BiLSTM-TL load factor prediction curve of small sample data

Table. 3: Comparison of predictions of various models with small sample data

models/evaluation merits	RMSE	MAE	MAPE
CNN	9.53	7.88	12.10
LSTM	9.42	8.27	10.48
CNN-LSTM	9.25	9.57	8.64
CNN-BiLSTM	8.55	6.76	7.99
CNN-LSTM-TL	6.53	4.54	5.72
CNN-BiLSTM-TL	6.11	4.82	5.47
CEEMADAN-LSTM	4.24	3.37	4.28
CEEMADAN-BiLSTM	3.97	2.98	3.63
CEEMADAN-BiLSTM-Attention	3.72	2.85	3.40

4.3. Discussion of model performance results

- i. Table. 2 shows that the hybrid models CEEMDAN-LSTM and CEEMDAN-BiLSTM based on CEEMDAN decomposition work better than some traditional machine learning methods CNN, LSTM and hybrid models (CNN-LSTM, CNN-BiLSTM) under the condition of sufficient data. Furthermore, the effect is further enhanced by adding Attention mechanism to improve the importance of the focused information of BiLSTM neural network. It can be seen that using CEEMDAN method to decompose the non-smooth and non-linear raw data can effectively extract the smooth information. Additionally, the accuracy of the forecast can be improved by utilizing this smooth information in the prediction process and superimposing the prediction findings.
- ii. As can be observed in Table. 3, this paper first employs migration strategies to construct two models CNN-LSTM-TL and CNN-BiLSTM-TL. And they outperform CNN, LSTM, CNN-LSTM, and CNN-BiLSTM models trained directly from the small sample data in terms of prediction results. However, the accuracy of predicting small sample data using the CEEMDAN-BiLSTM-Attention model still outperforms both CNN-LSTM-TL and CNN-BiLSTM-TL models using the migration

strategy. Although the first half of the validation set predictions are not as good as the CNN-BiLSTM-TL model according to the prediction curves. In general, conclusions can be drawn from the three evaluation indicators and trend of the prediction curve that the CEEMDAN-BiLSTM-Attention model can still be well applied to small data samples and achieve very good prediction accuracy.

- iii. The decomposition effect of CEEMDAN on the raw data causes the hybrid model run time of CEEMDAN to be generally longer than the other models mentioned above. This problem may be improved by using GPUs to accelerate the computation of neural networks.

5. CONCLUSIONS

Chiller load forecasting is of great importance for intelligent energy decision management of buildings and for developing energy distribution strategies for HVAC systems. In this paper, the CEEMDAN-BiLSTM-Attention method is applied to chiller load prediction. The original nonlinear and nonsmoothed data are first decomposed using CEEMDAN. For each distinct component of the decomposition, the prediction is then carried out using the BiLSTM neural network and Attention mechanism. Finally, the final prediction value is calculated by superimposing the predictions from various components. Experiments were conducted using data from two buildings, and by contrasting the prediction accuracy of various machine learning models to prove the method's efficiency. Meanwhile, for the problem of insufficient original data samples of new buildings, this paper compares migration strategy methods (CNN-LSTM-TL, CNN-BiLSTM-TL) and hybrid model methods based on CEEMDAN to conduct experiments on new buildings with lack of data, and the findings indicate that the CEEMDAN-BiLSTM-Attention model can also achieve good prediction results with small sample data, and the prediction accuracy is better than some migration models. For future work, the impact of different data sets on the method can be investigated, as well as considering the load prediction of other parts of the HVAC system, and combining the load of each part to achieve a rational distribution of energy.

REFERENCES:

- [1] S. Walker, W. Khan, K. Katic, W. Maassen, W. Zeiler, Accuracy of different machine learning algorithms and added-value of predicting aggregated-level energy performance of commercial buildings, *Energy and Buildings*. 209 (2020) 109705.
- [2] R. Cox, S. Walker, J. van der Velden, W. Zeiler, Chiller load forecasting using outdoor temperature profiles for a medium-sized office building, in: 2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), 2020, pp. 695-699.
- [3] R. Jing, M. Wang, R. Zhang, N. Li, Y. Zhao, A study on energy performance of 30 commercial office buildings in Hong Kong, *Energy and Buildings*. 144 (2017) 117-128.
- [4] L. Wang, EWM. Lee, RKK. Yuen, W. Feng, Cooling load forecasting-based predictive optimisation for chiller plants, *Energy and Buildings*. 198 (2019) 261-274.

- [5] A. C. Caminero, S. Ros, R. Hern, A. Robles-Gomez, R. Pastor, Load forecasting mechanism for e-learning infrastructures using exponential smoothing, in: 2011 IEEE 11th International Conference on Advanced Learning Technologies, 2011, pp. 364-365.
- [6] S. Singh, S. Hussain, M. A. Bazaz, Short term load forecasting using artificial neural network, in: 2017 Fourth International Conference on Image Information Processing (ICIIP), 2017, pp. 1-5.
- [7] X. Dong, L. Qian, L. Huang, A CNN based bagging learning approach to short-term load forecasting in smart grid, in: 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, 2017, pp. 1-6.
- [8] M. Imani, H. Ghassemanian, Sequence to image transform based convolutional neural network for load forecasting, 2019 27th Iranian Conference on Electrical Engineering (ICEE), 2019, pp. 1362-1366.
- [9] L. Ran, S. Fan, D. Xing, Ultra short-term load forecasting method for user-level integrated energy system considering spatio-temporal coupling of multi-energy, Power System Technology. 11 (2020) 4121-4134.
- [10] S. Luo, Y. Rao, J. Chen, W. Haijing, W. Zhao, Short-Term Load Forecasting Model of Distribution Transformer Based on CNN and LSTM, in: 2020 IEEE International Conference on High Voltage Engineering and Application (ICHVE), 2020, pp. 1-4.
- [11] D. Liu, Q. Yang, F. Yang, Predicting Building Energy Consumption by Time Series Model Based on Machine Learning and Empirical Mode Decomposition, in: 2020 5th IEEE International Conference on Big Data Analytics (ICBDA), 2020, pp. 145-150.
- [12] H. Zheng, J. Yuan, L. Chen, Short-term load forecasting using EMD-LSTM neural networks with a Xgboost algorithm for feature importance evaluation, Energies. 10.8 (2017) 1168.
- [13] D. Zhang, C. Cai, S. Chen, L. Ling, An improved genetic algorithm for optimizing ensemble empirical mode decomposition method, Systems Science & Control Engineering. 7.2 (2019) 53-63.
- [14] S. Gandham, B. Anuradha, An iterative method of Ensemble Empirical Mode Decomposition for enhanced ECG signal denoising, in: 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2016, pp. 1477-1480.
- [15] J. Cao, Z. Li, J. Li, Financial time series forecasting model based on CEEMDAN and LSTM, Physica A: Statistical mechanics and its applications. 519 (2019) 127-139.
- [16] X. Zhang, Y. Yang, Suspended sediment concentration forecast based on CEEMDAN-GRU model, Water Supply. 20.5 (2020) 1787-1798.
- [17] María E. Torres, Marcelo A. Colominas, G. Schlotthauer, P. Flandrin, A complete ensemble empirical mode decomposition with adaptive noise, in: 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP), 2011, pp. 4144-4147.
- [18] I. Karijadi, S. Y. Chou, A hybrid RF-LSTM based on CEEMDAN for improving the accuracy of building energy consumption prediction, Energy and Buildings. 259 (2022) 111908.
- [19] P. Zeng, C. Sheng, M. Jin, A learning framework based on weighted knowledge transfer for holiday load forecasting, Journal of Modern Power Systems and Clean Energy. 7.2 (2019) 329-339.
- [20] X. Fang, G. Gong, G. Li, L. Chun, W. Li, P. Peng, A hybrid deep transfer learning strategy for short term cross-building energy prediction, Energy. 215 (2021) 119208.
- [21] H. Yin, Z. Ou, J. Fu, Y. Cai, S. Chen, A. Meng, A novel transfer learning approach for wind power prediction based on a serio-parallel deep learning architecture, Energy. 234 (2021) 121271.
- [22] L. Feng, Z. Xu, Y. Xin, Load analysis and prediction of integrated energy distribution system based on deep learning, High Volt. 47 (2021) 23-32.
- [23] J. Koushik, Understanding convolutional neural networks, in: arXiv preprint, 2016, pp. 1605-09081.
- [24] L. Wenya, Cooling, Heating and Electric Load Forecasting for Integrated Energy Systems Based on CNN-LSTM, in: 2021 6th International Conference on Power and Renewable Energy (ICPRE), 2021, pp. 808-812.
- [25] Z. Shen, X. Fan, L. Zhang, H. Yu, Wind speed prediction of unmanned sailboat based on CNN and LSTM hybrid neural network, Ocean Engineering. 254 (2022) 111352.
- [26] C. C. J. Kuo, Understanding convolutional neural networks with a mathematical model, Journal of Visual Communication and Image Representation. 41 (2016) 406-413.
- [27] N. Grabsia, CEEMDAN And Fast Kurtogram Method For Bearing Fault Detection, in: 2019 6th International Conference on Image and Signal Processing and their Applications (ISPA), 2019, pp. 1-5.
- [28] L. Zhang, Word Sense Disambiguation Model based on Bi-LSTM, in: 2022 14th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 2022, pp. 848-851.
- [29] A. Graves, Long short-term memory, Supervised sequence labelling with recurrent neural networks. (2012) 37-45.
- [30] H. Cai, X. Chen, J. Ling, Q. Xu, Short-Term Load Forecasting Based on Radam Optimized CNN-BiLSTM-AE Hybrid Model, in: 2022 Power System and Green Energy Conference (PSGEC), 2022, pp. 626-631.
- [31] R. Barua, A. K. Sharma, Dynamic Black Litterman portfolios with views derived via CNN-BiLSTM predictions, Finance Research Letters. 49 (2022) 103111.
- [32] D. Laszuk, PyEMD documentation. (2020).
- [33] I. Kandel, M. Castelli, The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset, ICT express. 6.4 (2020) 312-315.