ISSN 2959-6157

Deepening Intelligent Microgrid Management: A Study on Improving Load Forecasting Accuracy Based on Informer Models

Yuke Wang

Sichuan Normal University, Chengdu, Sichuan, China

Abstract:

In the context of the "double carbon" strategy and the rapid development of deep learning, it provides new ideas for load forecasting of intelligent microgrids. In this study, we choose the Informer model based on the Transformer framework, which improves the self-attention mechanism and reduces the computational cost, to improve load accuracy and to achieve intelligent management of the microgrid system by accurately forecasting power load data.

Keywords: component, formatting, load forecasting, Informer, self-attention mechanism, microgrid

1. Introduction

With the aggravation of the global energy crisis and environmental pollution and the introduction of the "dual-carbon" strategy, the development and utilisation of renewable energy sources (e.g. solar, wind, etc.) has become a pivotal way to solve these problems. In terms of its classification as a novel type of power system architecture, smart microgrids show great potential for integrating renewable energy sources and improving grid flexibility and stability. However, the high level of uncertainty and volatility of renewable energy sources poses unprecedented challenges for microgrid management.

Load forecasting is a crucial aspect of smart microgrid management. Accurate forecasting of future electricity demand plays a core part in ensuring the security of supply, optimising resource allocation and reducing operating costs. Traditional load forecasting methods often fail to adequately capture the complex patterns and long-term dependencies hidden in the data, and thus may encounter problems such as insufficient accuracy or low robustness in practical applications.

As the global transition to a low-carbon economy advances, the integration of an increasing number of renewable energies into the existing energy system is becoming more and more prevalent. These energy sources, such as wind and solar, are characterised by intermittency and uncertainty, making traditional load forecasting models challenging. In recent years, deep learning techniques have made considerable advances in the area of time series forecasting and have been successfully applied to perform load forecasting in several scenarios. Meng(2024) et al. proposed an integrated energy forecasting model based on the combination of spatial temporal graph convolutional networks (STGCN) and the Transformer combined short-term load forecasting model for integrated energy systems[1]; Zhang(2023) et al. proposed a new model based on the combination of Transformer and graph convolutional networks (GCNs) for net load forecasting of electric power[2]; Sun(2023) et al. proposed a LSTM and multi-feature dynamic similarity day based integrated energy system load forecasting method, the objective is to utilise the advanced feature change law of the integrated energy system (IES) in order to enhance the precision of short-term load forecasting[3]; Yu (2022) et al. proposed a combined forecasting method based on chaos theory, variational modal decomposition VMD, integration of moving average autoregressive ARIMA model and gated recurrent unit GRU neural network in response to the problem of high stochasticity of short-term power loads and low prediction accuracy[4], Wang(2022) et al. proposed and established ARIMA-LSTM model by combining ARIMA model with LSTM model[5]. However, the above models, either single or combined models can only solve the short-term load forecasting, Zhou(2021) et al. verified that the LSTM model raises the MSE very high after 48h, and the LSTM model fails[6]; the various combined models also work on the forecasting accuracy, while ignoring the problems of forecasting length and computational cost.

Based on the above problems, in order to address the issues of weak prediction precision and short prediction accuracy and short prediction length of general models in prediction problems, and the high computational cost of Transformer, this study introduces the Informer model, which is based on the improved Self-Attention Mechanism under the framework of Transformer, and the Informer model, as a deep learning model with improved design based on the Self-Attention Mechanism, shows

great potential in many long-series time series prediction (LSTF) tasks. The Informer model, as a deep learning model based on the improved Self-Attention Mechanism, is more suitable for processing long series of time data and has shown great potential in many long series of time series forecasting (LSTF) tasks[6].

2. Random forest algorithm

Electricity load forecasting is a complex time-series problem that is influenced by a number of factors, such as historical load data, weather conditions, date type (weekday or holiday), seasonality, and trends. In order to enhance the prediction performance, feature selection is very critical. In this study, the Random Forest algorithm is used to analyse the relevance between multidimensional data and base electric load, screen out the feature parameters with high relevance, exclude the parameters with poor and irrelevant relevance, and reduce the complexity of data.

The Random Forest algorithm is an ensemble learning approach that enhances the overall model performance by constructing multiple decision trees and assembling their predictions. Feature selection using Random Forest better captures nonlinear patterns and is scalable enough to handle large-scale data and a large number of features, reducing the interference of weakly correlated features and thus improving the ability of the model to generalise[9]. The principle of Random Forest is based on two main concepts: integrated learning and decision trees.

(1) Integrated learning: integrated learning methods improve prediction performance by constructing and combining multiple models. Random forest is a typical example of applying this idea, which constructs multiple decision trees and integrates their results as a way to enhance the predictive accuracy and generalisation of a single model.

(2) Decision Tree: A decision tree is a simple predictive model represented as a tree structure where each internal node is represented by an attribute test, each branch is a test output, and each leaf node is a category label (decision result). The main challenge with decision trees is that they are prone to overfitting, especially when the tree is deep.

Feature selection using random forests mainly focuses on finding the features that have the highest correlation with the target variable and finding the features that express the optimal prediction with the smallest sample size[9].

3. Informer model

From Figure. 1, it can be seen that Transformer is able to process the training data in parallel, which is more efficient than the traditional recurrent neural network (RNN) and long short-term memory network (LSTM). Through the self-attention mechanism, the Transformer is able to directly establish a connection between any two positions in the input sequence, thus effectively solving the difficulty of capturing long-distance dependencies in RNN and LSTM.

However, despite the many advantages Transformer brings, it also has some disadvantages in practical applications. Transformer requires a lot of memory and computational resources, especially for very long input sequences. Because each layer involves a fully concatenated operation with complexity $O(L^2)$ (the length of the sequence), and the dynamic decoding operation of Transformer leads to a sudden drop in the output speed, the efficiency decreases drastically when it comes to sequences that are too long. Informer was born as a variant based on the Transformer framework designed for time series prediction.

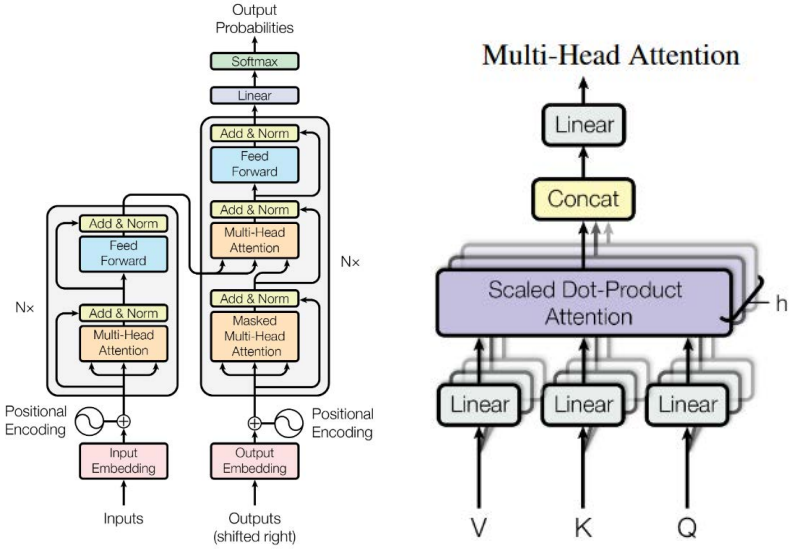

Figure 1 Structure of Transformer and its multi-head self-attention mechanism[9]

3.1 . *Structure of Informer*

The entire structure of Informer is illustrated in Figure. 2. To address the shortcomings of the Transformer model proposed above, the Informer is optimised in three ways:

1. more efficient processing of long-time sequences (ProbSparse self-attention mechanism)
2. Reduce space complexity (Self-attention Distilling mechanism)
3. Output all predicted values at once (Generative decoder mechanism)

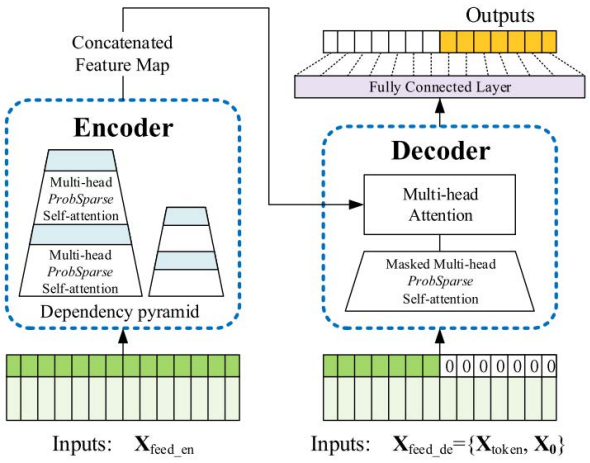

Figure 2 Structure of Informer[6]

Informer is based on the Transformer framework and improves on it, with the Encoder part on the left receiving extra-long input data (green part). The conventional Self-Attention layer is substituted with the ProbSpare Self-Attention layer introduced in this study. The blue part is the Self-Attention distilling process to perform feature condensation. The Encoder module enhances the robustness of the algorithm by overlaying the above two operations. On the right side is the Decoder section, which takes a series of long sequential inputs and pads the predicted goal position with 0. It measures the Self-Attention component on the Feature Map and then generates the predicted output (orange part).

3.2 *ProbSpare Self-Attention*

Informer introduces a ProbSparse self-attention mechanism, which decreases complexity by calculating only the key-value pairs that are most likely to have higher attention weights, thus enabling more efficient processing of long sequence data.

The traditional self-attention mechanism is based on a tuple input with the expression as

$$\mathrm{A}(Q,K,V)=Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

Q, K, V represent matrices of Query, Key, and Value, respectively, and are input dimensions

The probability form of the Attention coefficient for the ith query is

$$\mathrm{A}(q_i,K,V)=\sum_j\frac{k(q_i,k_j)}{\sum_l k(q_i,k_j)}v_j=E_{p(k_i|q_i)}\left[v_j\right] \tag{2}$$

$p(k_i\mid q_i)=\frac{k(q_i,k_j)}{\sum_l k(q_i,k_j)}$, $k(q_i,k_j)$ selects the asymmetric exponential kernel $\exp(\frac{q_ik_j^T}{\sqrt{d_k}})$, Attention mainly describes the relevance of query and key, and picks out the query and key with high relevance. As shown in Figure. 3, p is considered as some form of distribution, the closer it is to the uniform distribution, the less important the query ("Lazy" Query) is, and when the query is more active in some positions ("Active" Query), the weight difference is larger, the difference in weights is larger. The sparsity of the query is measured using the KL scatter measure, and the sparsity evaluation formula for the ith query is

$$M(q_i,K)=ln\sum_{j=1}^{L_K}e^{\frac{q_ik_j^T}{\sqrt{d}}}-\frac{1}{L_K}\sum_{j=1}^{L_K}\frac{q_ik_j^T}{\sqrt{d}} \tag{3}$$

The distribution of sparse self-attention is characterised by a long-tailed shape, whereby a small number of dot pairs contribute significantly to the main attention, while other pairs can be relatively neglected[7]. In the first half, all keys are represented by the log-sum-exp function (LSE), while in the second half, the arithmetic mean is calculated. The larger the value of the LSE, the more important it is for Attention and the more likely it is to be at the front of the long tail of the distribution. Based on the above analysis, the probabilistic sparse self-attention mechanism formula is obtained

$$\mathrm{A}(Q,K,V)=Softmax\left(\frac{\bar{Q}K^T}{\sqrt{d_k}}\right)V \tag{4}$$

It would be beneficial to ascertain the location of a sparse matrix with the same Q-width that contains only the largest $u=c\cdot lnL_Q$ queries under sparse evaluation, making the sparse self-attention mechanism only need to do O(lnL_Q) dot products in each query-key lookup. According to the above theory, it makes the sparse self-attention mechanism's overhead of each layer is reduced, and the computational complexity is reduced from O(L^2) to O(LlnL) [8]

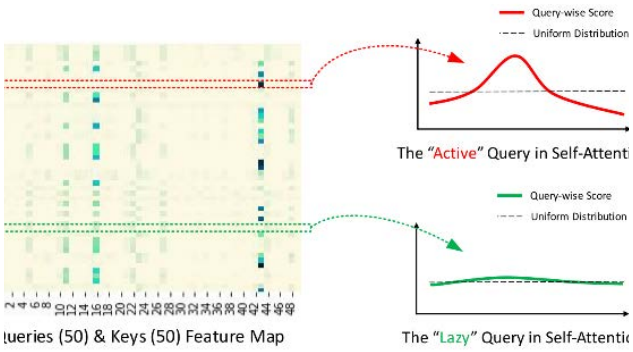

Figure.3 The more active the query in different positions in the self-attention mechanism the greater the difference in weights, and vice versa the smaller the difference

3.3 *Self-attention Distilling and Decoder*

In the Encoder module, the Informer uses Self-Attention Distilling, a distilling step that involves "distilling" the input sequence into shorter versions for use in subsequent layers. This means that the most informative or important parts of the original input data set are filtered out and other parts are removed to simplify the subsequent processing, reducing the memory and time required by the algorithm, as shown in Figure 4.

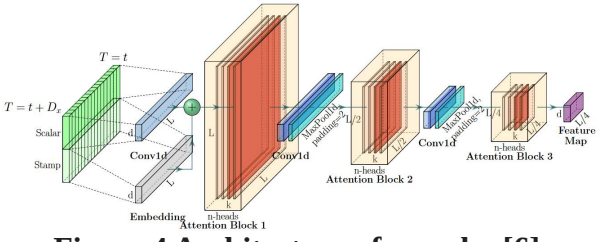

Figure 4 Architecture of encoder.[6]

The expression for the distillation operation from the jth layer to the j+1th layer is

$$X_j^t = MaxPool(ELU(Conv1d([X_j^t]_{AB}))) \tag{5}$$

$[X_j^t]_{AB}$ includes a multi-head probabilistic sparse self-attention mechanism and necessary procedures, and the output of each block passes sequentially through a Conv1d one-dimensional convolutional layer, an ELU activation layer, and a layer of maximum pooling with a stride length of 2 is appended after one layer is pile-up. To improve the resilience of the algorithm, a half-copy of the primary stack is built and the amount of self-attention distilling layers is progressively decreased by discarding one layer at a time so that their output dimensions are aligned. Finally, the outputs of all stacks are concatenated to yield the ultimate encoder.

The input time series X_{de}^t is divided into a sequence of historical loads X_{token}^t and a sequence of predicted loads X_0^t (where the scalar is filled with zeros) in the decoder's processing, which takes the following form

$$X_{de}^t = Concat(X_{token}^t, X_0^t) \in R^{(L_{token}+L_y)d_{model}} \tag{6}$$

The 100% interconnected connected layer generates the final output, generating all the predicted sequences at once to achieve the purpose of shortening the decoding time.

4. Evaluation indicators

this study assesses the predictive effectiveness of the model using mean absolute error (MAE) and mean square error (MSE), as defined by the following formulae.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{7}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{8}$$

The number of samples included in the training set is denoted by n; y_i、$\bar{y}$、$\hat{y}_i$ are the true value, the average of the true value and the predicted value of the data at time t, respectively. R^2 reflects the correlation between the true value and the predicted value, and the closer the value is to 1, it means that the model fits the model better; it reflects the degree of discrepancy between the estimated quantity and the quantity to be estimated, a measure; the MAE can avoid the problem of the mutual offset of the error, and thus the actual size of the prediction error can be accurately reflected. It can be observed that as the value of MSE and MAE increases, the prediction accuracy also increases. The lower the values of MSE and MAE, the greater the prediction accuracy.

5. Calculus analysis

5.1 . *Data description and data pre-processing*

In order to validate the veracity of this experiment, the standard dataset provided by the Ninth "China Electrical Engineering Society Cup" National University Students' Mathematical Modelling Competition for Electrical Engineering was used for the calculations [10]. The power load values and meteorological data (daily maximum temperature, daily minimum temperature, daily average temperature, daily relative humidity and daily rainfall) from 1 January 2012 to 10 January 2015 were selected for the purpose of predicting the short-term load of the power system in a certain region.

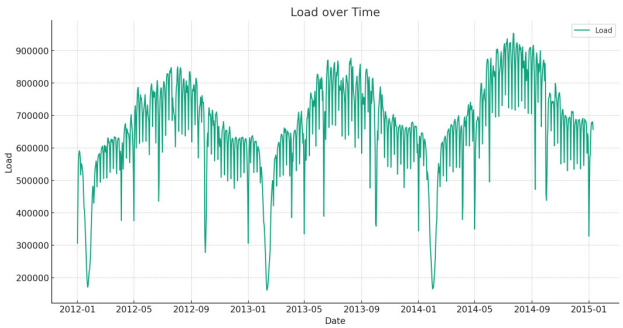

Figure 5 Load over Time

The total power load is shown in Figure. 5, which can be seen to present a cyclical-like feature, which may be affected by seasonal features. In this study, the features are divided into three categories, and the temperature feature with the largest width of change is decomposed into three to obtain six dimensions of input features, and then the six feature dimensions are analysed and screened.

5.2 . Feature processing and analysis

Using the random forest algorithm, the input data are screened for features, and the features with high relevance are selected to enhance the robustness of the prediction model; the feature components with low relevance are eliminated to avoid the overfitting phenomenon. In load forecasting, feature factors such as temperature, humidity and rainfall all have a certain impact on forecasting, and feature importance assessment is carried out by Random Forest and important features are selected, and the results are shown in Table 1 and Figure 7.

TABLE1. Characteristic factor correlation analysis

Feature Category	Feature	Correlation Coefficient
Temperature characteristics	Daily maximum temperature(℃)	0.137
	Daily minimum temperature(℃)	0.324
	Average daily temperature(℃)	0.359
Humidity characteristics	Daily relative humidity	0.126
Rainfall characteristics	Daily rainfall(mm)	0.053

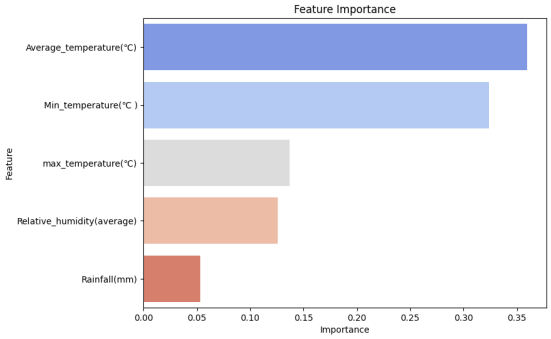

Figure. 6 Feature selection result graph

From Table 1 and Figure 6, it can be seen that the correlation of temperature factor is higher than humidity and rainfall, and according to the correlation coefficient, the correlation coefficients of factors such as temperature and humidity are in the range of 0.1 to 0.5, which is a strong correlation to the load. While the correlation coefficient of rainfall is lower than 0.1, the correlation is low, and the feature of rainfall is excluded in order to achieve the effect of reducing the calculation volume of the model.

TABLE2. Multivariate long series time series prediction results

Model	No Feature Extraction		Feature Extraction	
Metric	MSE	MAE	MSE	MAE
6	0.241	0.342	0.249	0.364
12	0.281	0.383	0.340	0.433
24	0.295	0.395	0.312	0.411
48	0.423	0.427	0.351	0.428
96	0.370	0.433	0.363	0.428

A comparison is made between the results of the integrated model prediction and the actual data, as shown in Figure 7, and from Table 2, it can be noticed that the model using feature extraction achieves better results in both MSE and

MAE metrics at longer time window lengths (48, 96). This indicates that Feature Extraction can help the model to better capture the key features in the input data, thus improving the prediction accuracy for longer time series. For time window lengths of 6, 12, and 24, the models without and with feature extraction have lower MSE and MAE, which implies that the models can predict loads more accurately over shorter time horizons.

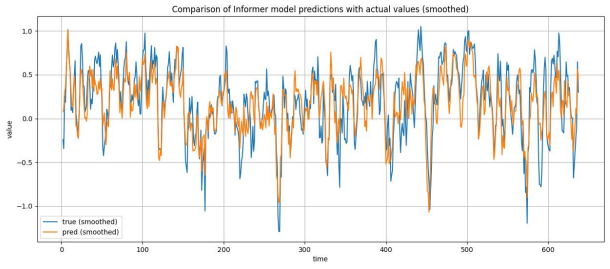

Figure 7 Comparison of real and predicted values of the Informer model

6. Conclusion

In this study, for the load forecasting problem, we propose a deferred model Informer based on the framework of Transformer's attention mechanism, study the influence of input features on the forecasting results, analyse the relevance of input features through the Random Forest Algorithm, retain the features with strong relevance, and improve the robustness of the forecasting model; eliminate the features with low relevance, reduce the dimension of inputs, and reduce the computational The features with low correlation are eliminated to reduce the dimensionality of the input, reduce the computational complexity, and avoid overfitting. Through experimental validation, it is concluded that although the performance of the model varies under different time window lengths, overall, the performance is relatively stable. This indicates that the Informer model is robust in dealing with the load prediction problem under different time scales; feature extraction can improve prediction accuracy for longer time series by helping the model to better capture key features in the input data.

References

[1] Meng W, Yu B, Bai L et al. Short-term electric net load forecasting based on STGCN-Transformer[J/OL]. China Test:1-9[2024-03-09].http://kns.cnki.net/kcms/detail/51.1714.TB.20240118.1534.002.html.

[2] Qingyong Zhang, Jiahua Chen, Gang Xiao, Shangyang He, Kunxiang Deng,TransformGraph: A novel short-term electricity net load forecasting model,Energy Reports,Volume 9,2023,Pages 2705-2717,ISSN 2352-4847,https://doi.org/10.1016/j.egyr.2023.01.050.

[3] Fan Sun, Yaojia Huo, Lei Fu, Huilan Liu, Xi Wang, Yiming Ma,Load-forecasting method for IES based on LSTM and dynamic similar days with multi-features,Global Energy Interconnection,Volume 6, Issue 3,2023,Pages 285-296,ISSN 2096-5117,https://doi.org/10.1016/j.gloei.2023.06.003.

[4] YU Junqi,NIE Guikai,ZHAO Anjun,et al. ARIMA-GRU short-term power load forecasting based on feature mining[J]. Journal of Power System and Automation,2022,34(03):91-99. DOI:10.19635/j.cnki.csu-epsa.000843.

[5] WANG Xin, LI Angui, LI Yang, et al. Integrated energy system load and wind resource forecasting based on ARIMA-LSTM model[J]. Journal of Xi'an University of Architecture and Technology(Natural Science Edition),2022,54(05):762-769. DOI:10.15986/j.1006-7930.2022.05.015.

[6] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 35(12), 11106-11115. https://doi.org/10.1609/aaai.v35i12.17325.

[7] LI Jiayi,ZHAO Bing,LIU Xuan,et al. Short-term load forecasting in station area based on DWT-Informer[J]. Electrical Measurement and Instrumentation,2024,61(03):160-166+191. DOI:10.19753/j.issn1001-1390.2024.03.022.

[8] Zhuomi Shi,Qiwu Ran,Fucong Xu. Short-term load forecasting based on aggregated quadratic modal decomposition and Informer[J/OL]. Grid Technology:1-15[2024-04-16].https://doi.org/10.13335/j.1000-3673.pst.2023.1467.

[9] Vaswani A , Shazeer N , Parmar N ,et al.Attention Is All You Need[J].arXiv, 2017.DOI:10.48550/arXiv.1706.03762.

[10] Electromathematics Committee of China Electrical Engineering Society. The ninth "Chinese electrical engineering society cup" national college students electrical mathematical modelling competition title [EB/OL]. (2016-04-25) [2018-05-10]. http://shumo. nedu. edu. cn.