Short-term power load forecasting using informer encoder and bi-directional LSTM

Shiyu Tan*, Yuhao Yang, and Yongxin Zhang
Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering, China
Three Gorges University, Yichang, Hubei, 443002, China

Abstract. An innovative model called InE-BiLSTM is proposed here, which combines the Informer Encoder with a bidirectional LSTM (Bi-LSTM) network. The goal is to enhance the precision and efficacy of short-term electricity load forecasting. By integrating the long-term dependency capturing capability of the informer encoder with the advantages of Bi-LSTM in handling dynamic features in time series data, the InE-BiLSTM model effectively addresses complex patterns and fluctuations in electricity load data. The study begins by analyzing the current state of short-term electricity load forecasting, followed by a detailed introduction to the structure and principles of the InE-BiLSTM model. Results of the experiment demonstrate that, compared to the Informer, traditional Bi-LSTM, and Transformer models, the InE-BiLSTM model consistently outperforms them across various evaluation metrics.

Keywords. Short-term power load forecasting, InE-BiLSTM model, Informer encoder, Bi-LSTM.

1 Introduction

As global energy demand continues to rise and the share of renewable energy in energy mix gradually increases, the operation and management of power systems face unprecedented challenges and opportunities. Against this backdrop, load forecasting has emerged as an indispensable cornerstone in the planning, operation, and market transactions of power systems. Among them, short-term electricity forecasting holds a unique position. Accurate short-term load forecasting can help power system operators optimize generation schedules, manage reserve capacity, reduce operational costs, and enhance system reliability and efficiency. It bears pivotal importance in safeguarding the economic stability of the power grid.

Furthermore, the rapid development and evolution of the electrical network also demand continuous improvement of short-term load forecasting models to adapt to new changes. Factors such as the volatility of renewable energy sources and the smart grid pose new requirements for model design and optimization. Therefore, it is of paramount theoretical and practical significance to explore more efficient, accurate, and adaptable methods for short-term load forecasting.

* Corresponding author: 1923295312@qq.com

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).
2 Related work

In the past few decades, the field of power load forecasting has evolved from conventional statistical models to deep learning frameworks. In the early stages of short-term electricity load forecasting, traditional statistical techniques like time series analysis and regression analysis were commonly used [1][2]. These methods included models like the ARIMA, exponential smoothing, seasonal decomposition of time series (STLF), and the grey model proposed by Chinese scholar Professor Deng Julong [3]. These models analyze historical data to establish the relationship between power load and time for future load forecasting. However, these methods have certain limitations in handling nonlinear relationships, time variability, and complex dynamic power load data.

In the preceding years, the rapid achievement of deep learning has brought new breakthroughs to short-term power load forecasting. CNN, RNN [4] and their variants such as LSTM networks [5] and GRUs [6], as well as LSTM models with focus mechanism [7], are widely applied in short-term power load forecasting due to their powerful feature extraction capabilities and ability to model complex data relationships. In addition, the Transformer models originally used for Natural Language Processing tasks have gradually found applications in time series prediction and surprisingly achieved excellent results [8]. The Informer model, proposed in 2020 [9], further optimized the efficiency issues of the Transformer in time series prediction. It combines and enhances the Transformer's self-attention mechanism and modeling capabilities for long and short-term dependencies, effectively addressing challenges posed by longer sequence lengths and multiple scale features in power load forecasting. This pioneering methodology has led to a notable enhancement in the efficacy of time series forecasting models.

Indeed, researchers’ focus extends beyond individual models. A CNN models combining the transformer is proposed to enhance the overall model's ability to extract contextual information in [10], building upon the foundation of the Transformer model. Following that, researchers proposed combining LSTM models with Transformer models, utilizing LSTM’s sequence modeling capabilities to preprocess input sequences, and then leveraging Transformers to mitigate the model’s tendency to forget sequence information [11]. The FEDformer model proposed in [12] is also based on improvements to the Transformer model and demonstrates greater efficiency in long-term time series forecasting tasks. [13] introduced VMD-BEGA-LSTM (VLG) as a novel forecasting system for short-term electric load prediction, demonstrating its applicational value. Furthermore, reinforcement learning methodologies have been harnessed in electric load forecasting to attain superior predictive outcomes [14].

Building upon these technological advancements, this paper introduces a novel fusion model, InE-BiLSTM, combining Informer Encoder and Bi-LSTM structures to optimize electricity load forecasting. Initially, the time series is preprocessed through the Informer Encoder to effectively extract long-term features; subsequently, Bi-LSTM is utilized for in-depth temporal analysis, enhancing the model's understanding of time series [15]. By integrating these structures, the InE-BiLSTM model not only strengthens its capability to process sequence data but also captures complex patterns within the time series, significantly improving performance in short-term forecasting tasks. Experimental results demonstrate that InE-BiLSTM outperforms the standalone Informer model, traditional Bi-LSTM, and other baseline models across multiple forecasting tasks, proving its effectiveness and superiority in short-term time series prediction.
3 Model building process

3.1 Informer Encoder

The Informer Encoder consists of multiple encoder layers, followed by a downsampling layer and a normalization layer. Specifically, each encoder layer consists of three sub-layers: multi-head sparse self-attention, a feed-forward network, and residual connections. The utilization of the multi-head self-attention block enables the model to acquire learning different relationships in various projection spaces, thereby enhancing the model's abstract representation capability for positions across the sequence. The feed-forward network processes the self-attention block's output through non-linear mapping, further improving the model's expressive power. Residual connections contribute to mitigating the vanishing gradient problem, ensuring smooth information flow.

Fig. 1. Informer encoder model.

3.1.1 ProbSparse self-attention

In the calculation of attention, many queries may not provide effective values, thus, in ProbSparse self-attention, only positive queries are selected to compute the attention mechanism. In other words, this mechanism introduces probabilistic sparsity by selectively focusing on a subset of positions in the sequence, thereby reducing computational complexity. The approximate sparsity measurement for the $i$ query is defined as follows:

$$M(q_i, K) = \max(q_i^T K) - \frac{1}{L_k} \sum_{j=1}^{L_k} q_i^T k_j^T$$  \hspace{1cm} (1)

It calculates $M(q_i, K)$ by sampling randomly chosen $U = L_q \ln L_k$ dot product pairs, obtaining a sparsity measurement score for each query. As the score increases, the query is perceived in a more favorable light. Subsequently, based on a predefined parameter $u$, the top $u$ most positive queries are selected. Here, $d$ represents the input dimension.

Therefore, we permit each key to focus on only $u$ of the most positive queries to obtain ProbSparse self-attention. This leads to the formulation below:

$$u = c \cdot \ln L_q$$  \hspace{1cm} (2)
\[ \text{ProbAtt}(Q, K, V) = \text{Softmax} \left( \frac{Q K^T}{\sqrt{d}} \right) V \]  

where \( c \) represents an introduced sampling factor. In equation (2), \( \tilde{Q} \) is a sparse matrix of the same size as \( q \) and exclusively contains the queries that have been selected after the top-\( u \) filtration. \( K \) and \( V \) refers to matrices composed of key vectors and value vectors.

### 3.1.2 Multi-head Self-Attention

The model's capacity to comprehend input sequences is greatly improved through the utilization of the multi-head attention block and learning in parallel. Each head maps the input sequence to a different projection space through a linear transformation and then computes attention weights. By dividing the model into multiple heads, distinct subspaces are formed, enabling the model to focus on diverse aspects of information. The computational formula is as follows:

\[ Q_i = Q W_i^\tilde{Q}, K_i = K W_i^K, V_i = V W_i^V \]  

\[ \text{head}_i = \text{ProbAtt}(Q, K, V) \]  

\[ \text{MultiH}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h) W_o \]

where \( W_i^\tilde{Q} \in \mathbb{R}^{D \times d_k}, W_i^K \in \mathbb{R}^{D \times d_k}, W_i^V \in \mathbb{R}^{D \times d_v}, W_o \in \mathbb{R}^{h \times d_v} \), and there \( W_i^\tilde{Q}, W_i^K, W_i^V \) represents the parameter matrix, \( W_o \) represents the weight matrix for queries, and \( d_k, d_v \) represent the weight matrices for keys and values, respectively. Additionally, \( d_k = d_v = D / h \). The collective output of each head is combined before being forwarded to a linear layer to undergo subsequent processing.

### 3.1.3 Distilling layer

Within the feature mappings of the encoder, there exist redundancies in combined values. The utilization of distillation operations serves to accentuate key features with primary significance, assigning them higher weights. In the subsequent layer, these highlighted features will be employed to generate more focused self-attention feature mappings. The transition between \( j \)-th and the \( j+1 \)-th tier occurs as distilling moves forward.

\[ X'_{j+1} = \text{MaxPooling}(\text{ELU}(\text{Conv1d}[X'_j])) \]

\([\cdot]_{d_h} \) encapsulates the essential functions of the attention block, \( \text{Conv1d}() \) denotes a convolution operation in one dimension performed on temporal data sequences, employing the Exponential Linear Unit (ELU) as an activation function. Following the addition of such a layer, a max-pooling layer is included to effectively minimize memory usage by subsampling the temporal data.

### 3.2 Bi-LSTM layer

The structure of Bi-LSTM consists of two directions of LSTM layers, namely the forward LSTM and the backward LSTM. Each LSTM layer comprises input gates, forget gates, output gates, and memory cells, which controls the flow and retention of information. While the forward LSTM handles the input sequence chronologically, its counterpart, the
backward LSTM, tackles it in reverse. At each time step, the outputs from both LSTM directions converge to shape the ultimate output of the Bi-LSTM.

3.3 Proposed method: the InEn-BiLSTM

As depicted in the Figure 2, InEn-BiLSTM model integrates the encoder of informer with a Bi-LSTM module, aiming at enhancing the accuracy of electric load forecasting by leveraging the self-attention block of Informer and the bidirectional information processing capability of Bi-LSTM. Initially, the Informer Encoder conducts deep feature extraction on the input sequence using techniques such as ProbSparse Self-Attention, fully connected feedforward networks, and temporal feature aggregation, enhancing the capture of global and local relationships in long sequence data. Subsequently, the Bi-LSTM layer undertakes bidirectional temporal modeling, effectively understanding the time dependencies within the sequence by integrating past and future information to bolster time series analysis. Finally, features from both the informer encoder and Bi-LSTM layer are integrated through a fusion layer to produce accurate forecasting results.

Fig. 2. InEn-BiLSTM Model.

4 Experiment

4.1 Experimental environment

In this experiment, we utilized an experimental server configured with PyTorch 1.11.0, Python 3.8 (running on Ubuntu 20.04), and CUDA 11.3. The hardware environment comprises an NVIDIA RTX 4090 GPU with 24GB of memory, an Intel Xeon Platinum 8352V CPU boasting 12 cores, and a total system memory of 90GB. In terms of storage, the system disk has a capacity of 30GB.

4.2 Description and preparation of data

In this study, we selected the total electricity consumption data of all users in the Yichang city power system, aiming for a more accurate electricity load forecasting. The time span covers from December 2022 to August 2023. The dataset comprises the total electricity consumption information for 5000 users, sampled at a frequency of 15 minutes, resulting in a total of 26,209 data points.
During the preprocessing stage, several steps were implemented to ensure data quality and consistency. These steps included handling missing values, data normalization, and outlier processing. The dataset underwent imputation for missing values using linear interpolation to maintain the continuity of the time series. Additionally, as shown in equation (8), the Interquartile Range (IQR) method was employed to identify and remove outliers, minimizing the impact of noise on model training. To eliminate the influence of different scales, all features were standardized in this study, ensuring a dataset with a mean centered at 0 and a standard deviation fixed at 1.

\[ \text{IQR} = Q_3 - Q_1 \]  

In the equation (8), \( Q_1 \) represents the first quartile, \( Q_3 \) and represents the third quartile.

Finally, apportion the meticulously curated dataset into segments for training, validation, and testing, allocating 80%, 10%, and 10% respectively.

This dataset was applied to the task of electricity load forecasting, aiming to optimize the energy dispatching of the power system. The strategies employed during preprocessing contribute to both training efficiency and prediction accuracy of the model.

4.3 Parameterization

The experimental hyperparameter configurations are outlined in Table 1. The foundational model parameter configurations are detailed in Table 2. In designing the parameters, the experiment took into consideration a balance between the model's learning capacity, computational efficiency, and generalization ability. The selection of appropriate learning rates and batch sizes is crucial to ensuring the stability and efficiency of model training. Smaller learning rates and moderately-sized batches contribute to stable convergence during the training process while reducing memory consumption.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Rate</td>
<td>0.0002</td>
</tr>
<tr>
<td>Dropout</td>
<td>0.001</td>
</tr>
<tr>
<td>Patience</td>
<td>4</td>
</tr>
<tr>
<td>Batchsize</td>
<td>64</td>
</tr>
<tr>
<td>Epoch Number</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 1. Hyperparameter settings.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input_dim</td>
<td>Input dimension for the proposed model:1</td>
</tr>
<tr>
<td>Encoder layers</td>
<td>Layer of Informer Encoder:2</td>
</tr>
<tr>
<td>BiLSTM_input</td>
<td>Input dimension for the BiLSTM layer:512</td>
</tr>
<tr>
<td>BiLSTM_hidden_dim</td>
<td>Hidden dimension for the BiLSTM layer:256</td>
</tr>
<tr>
<td>Heads</td>
<td>Number of heads:8</td>
</tr>
<tr>
<td>Output_size</td>
<td>Out dimension for the proposed model:1</td>
</tr>
<tr>
<td>Factor</td>
<td>( \text{ProbSparse} ) attention factor:5</td>
</tr>
</tbody>
</table>

Table 2. InEn-BiLSTM’s main parameter configuration.

In addition, we employed the Adam optimizer in the experiments, which has demonstrated outstanding performance in handling time series forecasting tasks.
4.4 Interpretation of result

4.4.1 Assessment metrics

In this paper, we utilized various assessment metrics to thoroughly evaluate the effectiveness the InEn-BiLSTM model in the forecasting task. These evaluation metrics encompass MSE, RMSE, MAE, MAPE, and MSPE. By employing these multidimensional evaluation metrics, we can scrutinize the predictive precision and consistency of the model from different perspectives, providing a robust foundation for the holistic assessment of model performance.

4.4.2 Analysis and comparison

In the experiment, we configured our prediction horizon to be 12 test points, corresponding to a time interval of 3 hours for forecasting electricity load conditions. To facilitate comparison and observation, experiment employed normalization techniques in both plotting and error computation. This normalization approach provides a more intuitive sense of the accuracy of predictions.

Figure 3 displays the overall prediction results of the proposed model, demonstrating a remarkably close alignment with the actual outcomes.

Fig. 3. Integral load prediction.

To enhance model performance, we conducted experiments illustrated in Figure 4 to determine the optimal value for the "patience" parameter. Early Stopping is an effective method to prevent model overfitting, deciding when to halt training by monitoring the model's performance on a validation set. The results indicate that the model performs best when patience is set to 4. Hence, we select this value for subsequent experiments. Early Stopping, by timely terminating the training process, not only saves computational resources but also improves the model's predictive accuracy on unseen data.

Fig. 4. The efficacy of the suggested model under various patience.
To emphasize the superiority of the InEn-BiLSTM model, the experiment chose to compare it with two foundational models: Informer and Bi-LSTM. Additionally, the experiment also included the Transformer model as a reference. Figure 5 illustrates the predictive performance of these four models on randomly sampled data sample. Notably, the InEn-BiLSTM model closely aligns with the actual load curve. In the comprehensive prediction experiments, their comparative errors are summarized in Table 3. Furthermore, Figure 6 focuses on visualizing the MSE, MAE, and RMSE, with the proposed model represented by blue bars.

![Fig. 5. Comparison of the same data sample.](image)

**Table 3. Comparison of model's evaluation metrics.**

<table>
<thead>
<tr>
<th>Models</th>
<th>InEn-BiLSTM</th>
<th>Informer</th>
<th>Bi-LSTM</th>
<th>Transformer</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAE</td>
<td>0.2278</td>
<td>0.2512</td>
<td>0.4236</td>
<td>0.5196</td>
</tr>
<tr>
<td>MSE</td>
<td>0.0860</td>
<td>0.1058</td>
<td>0.2910</td>
<td>0.4080</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.2933</td>
<td>0.3254</td>
<td>0.5394</td>
<td>0.6388</td>
</tr>
<tr>
<td>MAPE</td>
<td>2.5804%</td>
<td>3.1146%</td>
<td>3.6257%</td>
<td>5.6651%</td>
</tr>
<tr>
<td>MSPE</td>
<td>890.2976%</td>
<td>155.0575%</td>
<td>1202.9364</td>
<td>2741.2461%</td>
</tr>
</tbody>
</table>

![Fig. 6. Columnar comparison chart of model's evaluation metrics.](image)
From the comparative experiments mentioned above, it is evident that the Informer model’s introduction of a sparse attention mechanism enables it to handle temporal dependencies and missing values more effectively than both the Bi-LSTM and Transformer models. While the Transformer model suffers from computational inefficiency and sensitivity to input sequences, it also lacks robust modeling of the temporal dimension. Consequently, it performs least favorably among the evaluated models. In contrast, the InEn-BiLSTM architecture first extracts sequence features using the Informer Encoder and subsequently feeds these features into the Bi-LSTM for bidirectional sequence scanning and further dependency extraction. As a result, the InEn-BiLSTM model exhibits significant advantages in short-term predictions and consistently outperforms other models across various evaluation metrics, achieving more accurate forecasting results.

In addition, to validate the effectiveness of each module in the InE-BiLSTM model, this study also conducted comparative experiment in Figure 7. For ease of explanation, experiment define Model1 as the InE-BiLSTM model with the traditional attention block replacing the ProbSparse self-attention block in the Informer Encoder. Model2 represents the InE-BiLSTM model with the distilling layer removed, while Model3 corresponds to the InE-BiLSTM model with the multi-head self-attention block replaced by a single-head self-attention block.

![Fig. 7. Predicted performance after removing different modules: (a) Prediction result of InEn-BiLSTM model, (b) Prediction result of model1, (c) Prediction result of model2, (d) Prediction result of model3.](image)

It is evident that each module of the InEn-BiLSTM model has contributed to its performance in this prediction task, they are all indispensable components.
5 Conclusion

This treatise proposes an original model, InEn-BiLSTM, designed to address the challenges in short-term electricity load forecasting. The model ingeniously combines the sequence feature capturing capability of Informer Encoder with the advantage of handling temporal dynamics in Bi-LSTM, effectively extracting and utilizing complex patterns in electricity load data. By introducing Informer's sparse self-attention block, the model achieves efficient grasp the correlation of prolonged sequences in time series while maintaining lower time complexity. Furthermore, through the integration of the Bi-LSTM module, InEn-BiLSTM enhances the understanding of short-term dependencies, enabling more accurate predictions of short-term variations in electricity load. Experimental results demonstrate that the InEn-BiLSTM model exhibits significant performance advantages over informer, traditional Bi-LSTM, and Transformer models in short-term electricity load forecasting tasks.

References