Multistep Forecasting Method of Short-term Power Load Based on VMD-Prophet-Seq2seq

Xingchen Song, Tongxi Wang, Ziming Xu, Zhan Shi

Abstract—**Power load forecasting is becoming increasingly important as power system management on offshore platforms flourishes in modernization. When forecasting the power load, increasing the forecasting step causes the forecasting error to increase. This paper proposes the VMD-Prophet-Seq2seq model, which performs multistep prediction on single-feature original data. First, the model uses the variational modal decomposition (VMD) method, which decomposes the original power load sequence into multiple integral mode functions (IMFs). Correspondingly, the Prophet method decomposes the original power load series into subsequences such as 'trend' and 'daily'. We process each IMF and subsequence separately into independent supervised-type data. Then, we encode each supervisor-type data with Seq2seq's encoder and output a corresponding number of subcoding vectors. Finally, the decoder sums these subcoding vectors and decodes the summed coding vectors to obtain the forecasting sequence. The experimental results show that the mean absolute percentage error (MAPE) value is at most 1.78% lower than that of the suboptimal model for a prediction length of 30.**

Index Terms—**Power Load, Forecasting, Variational Modal Decomposition, Prophet, Seq2seq**

I. INTRODUCTION

FORECASTING the power load helps adjust the power planning of offshore oil and gas field platforms. Power load data are collected and recorded at time intervals. Studying time-series forecasting is also helpful for power load forecasting. L. Suganthi et al. proposed statistical models for time-series forecasting [1]. On this basis, Kumar S. Vasantha and Siti Normah Hassan et al. studied a time-series forecasting method that combines the autoregressive integrated moving average (ARIMA) model with seasonal factors [2],[3]. They found that combining seasonal factors can reduce forecast error. Statistical models such as ARIMA methods require that the data be stationary. The prediction error is large when time series appear as a nonstationary stochastic process. Without combining seasonal factors, statistical models have limited ability to forecast nonstationary stochastic time series. Yildiz Baran and Elena Maria studied various machine learning (ML) models to forecast nonstationary stochastic time-series data [4],[5]. They showed the ML models are more suitable than statistical and analytical models for forecasting nonstationary stochastic time-series data. Meanwhile, Mosavi, Amir et al. proposed that hybridization, data decomposition, algorithm ensembles, and model optimization effectively improve the forecast accuracy of ML models [6]. Jiang Ping, Nguyen Linh and Sadaei Hossein Javedani et al. provided some combined models of ML combined with multiobjective optimization methods, ML combined with seasonal factors, and ML combined with statistical models [7],[9]. The research analysis of the above ML model shows that combined models of ML have better predictive outcomes than individual ML models. ML models require feature engineering and model tuning to reduce forecast errors. However, feature engineering is time-consuming and relies on expert experience. A review by Sezer, OB et al. suggested that DL models learn significantly better than ML models [10]. In contrast, deep learning (DL) models rely less on feature engineering and expert experience and emphasize network design and training strategies. Jihoon Moon et al. demonstrated that an artificial neural network (ANN) is more predictively accurate than supply vector regression (SVR) in time-series forecasting [11]. Xiang Zhongrun et al. used sequences to make time-series forecasts for sequence models and long short-term memory (LSTM) with accuracy [12]. The results prove that adjusting the external structure of the model can improve the accuracy of time-series forecasting. Bukhari Ayaz Hussain, Brian S. Freeman, and Li Taoying et al. proposed combining the LSTM and autoregressive fractionally integrated moving average (ARFIMA), decision tree, and convolutional neural network (CNN) forecast models, respectively. These combinatorial models showed better predictive effects than LSTM [13],[15]. The above study shows that deep learning models tend to exhibit better forecasting accuracy than statistical and machine learning models in time-series forecasting. At the same time, the single model is combined with other feature engineering or other models to improve the accuracy of forecasts.

In this paper, power load data are affected by daily load changes, temperature changes, or holidays, and the power load data have a certain periodicity. We start by separating the periodicity of the data and combining the separation method and the DL model. The primary aim of this research is to establish a suitable and accurate data separation model.

Manuscript received January 26, 2022; revised May 22, 2022.

This work is supported by the National Natural Science Foundation of China under Grant (No. 61703278).

Xingchen Song is a postgraduate student of Yangtze University, Jingzhou, 434000 China (e-mail: 201971332@yangtzeu.edu.cn).

Tongxi Wang is a professor of Software Engineering Department, Yangtze University, 434000 China (corresponding author; e-mail: txwang@yangtzeu.edu.cn)

Ziming Xu is a postgraduate student of Yangtze University, Jingzhou, 434000 China (e-mail: 202071547@yangtzeu.edu.cn).

Zhan Shi is a postgraduate student of Yangtze University, Jingzhou, 434000 China (e-mail: 202071530@yangtzeu.edu.cn).

II. RELATED WORK

A. Time Series Decomposition Method

The Fourier transform (FT) method is the most applied classical signal processing method, but an FT can only deal with unstable time-series. Later, J. Morlet proposed using a wavelet transform (WT). He replaced the infinite-length trigonometric basis function with a decaying wavelet basis function. This makes the WT more suitable for nonstationary processing data than the FT, but the wavelet basis function needs to be artificially selected. NE Huang, Z Shen, and SR Long et al. believed that the empirical mode decomposition (EMD) method can better handle nonstationary data and does not require manual parameter settings [16]. X. Qiu, X. Kong, and Y. Wei, et al. decomposed time-series data using the EMD and ensemble empirical mode decomposition(EEMD) methods [17],[19]. However, EMD cannot control the number of decomposition sequences. Dragomiretskiy K et al. proposed replacing the EMD method with the variational modal decomposition (VMD) method, thus solving the disadvantage of not decomposing a fixed number of sequences [20]. Gyamerah et al. used a combination of EMD and VMD with a generalized additive model(GAM) to forecast bitcoin prices [21]. The authors found that the forecasting error of the combined VMD method was smaller than that of the combined EMD method. In addition, Facebook proposed the Prophet method based on the STL [22] method, which can decompose the periodicity of time series. Prophet methods can decompose time series into seasonality, trends, and add-ons. We can comprehensively decompose time-series data using signal decomposition and prophetic methods.

B. Neural Network Methods

The right forecasting strategy can reduce the error of time-series forecasts. The multiple output strategy (MOS) is the most widely used multistep forecasting method among the multistep forecasting strategies [23]. The MOS requires fitting the dependencies between multiple inputs and multiple outputs using a model with more forecasting power. The ML model fits better than the statistical algorithm and has better forecasting capability for nonlinear series [24]. H. Hewamalage, Rahman Aowabin, and Shahid Farah et al. used a recurrent neural network (RNN) to forecast long sequences, showing better forecasting results than statistical models [25],[27]. However, single DL models cannot achieve satisfactory forecasting results in forecasting the power load data of offshore oil and gas platforms. Therefore, in this paper, the forecasting error in multiple steps is reduced based on the existing DL models combined with data decomposition methods.

C. Model Design

Our model uses the time-series decomposition method to decompose the intrinsic factors in the original data. We then perform feature extraction on these factors. We need to consider how to accurately breakdown the intrinsic factors in the raw data. The VMD and Prophet methods can help us decompose the power load data. The VMD method is an adaptive signal processing method that can fix the number of decompositions. It also solves the problems of mode mixing and end effects. The VMD method can decompose a complex time series into several stationary subsequences with different frequency scales. Solving the variational problem in the VMD method requires that the sum of the individual integral mode functions (IMFs) equals the original signal. Moreover, variational optimization needs to consider band limits and inherent multivariate modulation oscillations [28]. The variational constraint expression is shown in (1):

$$\min_{\{\mathrm{U_k}\},\{\omega_k\}}\{\sum_k\left\|\partial_t[(\delta(t)+\frac{\mathrm{j}}{\pi t})*u_k(t)]e^{-\mathrm{j}\omega_k t}\right\|_2^2\} \\ \cdot s.t.\sum_k \mathrm{u_k}=f \tag{1}$$

where K is the number of IMFs, $\{\mathrm{u}_k\}\{\omega_k\}$ is the central frequency after decomposition, $\delta(\mathrm{t})$ is the Dirac function and $*$ is the convolution operator. Converting constrained variational problems to unconstrained variational problems requires the introduction of a Lagrange multiplier λ. This calculation process is shown in (2).

$$\begin{aligned} &L(\{u_{lk}\},\{w_k\},\lambda)= \\ &\cdot\alpha\sum_k\left\|\partial_t[(\delta(t)+\frac{\mathrm{j}}{\pi t})*u_k(t)]e^{-\mathrm{j}\omega_k t}\right\|_2^2 \\ &\cdot+\left\|f(t)-\sum_k u_k(t)\right\|_2^2+\left\langle\lambda(t),f(t)-\sum_k U_k(t)\right\rangle, \end{aligned} \tag{2}$$

where α is the quadratic penalty factor. Solving (2) yields the modal component u_k and the central frequency ω_k. The solution process is expressed as follows (3).

$$\begin{aligned} &\hat{u}_k^{n+1}(\omega)=\frac{\hat{f}(\omega)-\sum_{i\neq k}\hat{u}_i(\omega)+\frac{\hat{\lambda}(\omega)}{2}}{1+2\alpha(\omega-\omega_k)^2}, \\ &\omega_{\mathrm{k}}^{\mathrm{n+1}}=\frac{\int_0^\infty\omega|\hat{u}_k(\omega)|^2d\omega}{\int_0^\infty|\hat{u}_k(\omega)|^2d\omega}. \end{aligned} \tag{3}$$

The Prophet method yields four series, trend $g(t)$, period $s(t)$, holiday $h(t)$, and residual ε_t, by decomposing the periodicity of the original time series. The $g(t)$ in the Prophet method is based on the logistic function and the piecewise linear function. $g(t)$ is as follows (4).

$$\mathrm{g}(t)=\frac{C(t)}{\exp(-(k+a(t)_t\delta)\cdot(t-(m+a(t)^T\gamma)))+1}, \tag{4}$$

where $a(t)$ is the logistic regression function and δ denotes the amount of variation in the growth rate at time t. The periodic term usually shows a specific variation with days, weeks, months, etc. The Fourier series representation of $s(t)$ takes the form (5).

$$\mathrm{s}(t)=\sum_{n=1}^N(a_n\cos(\frac{2\pi nt}{P})+b_n\sin(\frac{2\pi nt}{P})), \tag{5}$$

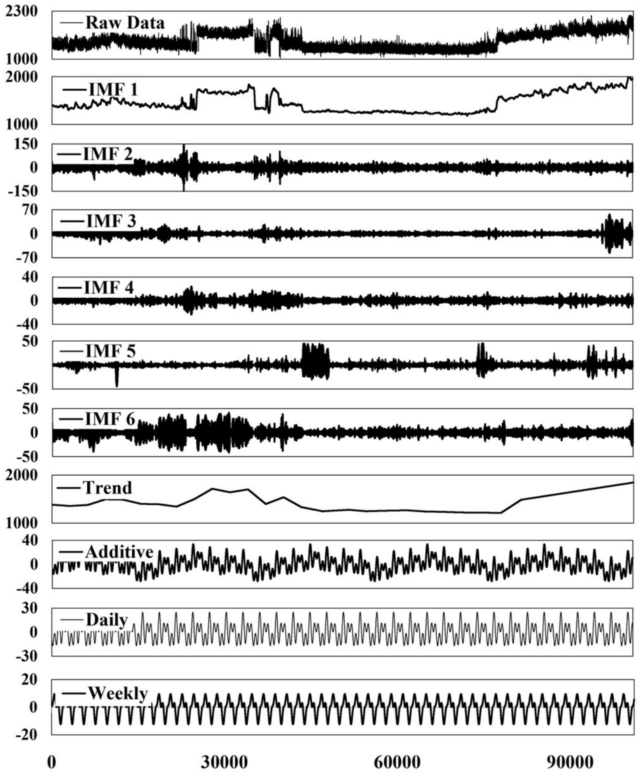
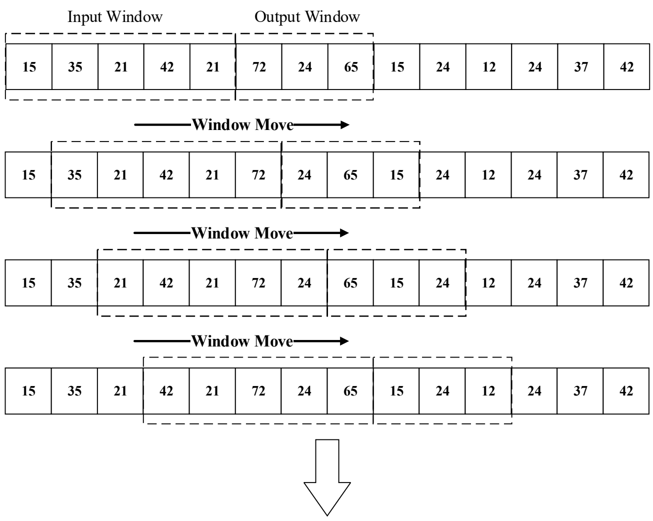


Fig. 1. Decompose the sequence of intrinsic factors from the preprocessed time-series data.



Input					Output		
15	35	21	42	21	72	24	65
35	21	42	21	72	24	65	15
21	42	21	72	24	65	15	24
42	21	72	24	65	15	24	12

Fig. 2. Converting data to supervised learning data.

where P denotes the period series, the model uses the indicator function to denote the holiday term, and $h(t)$ is expressed as shown in (6).

$$h(t)=Z(t)k=\sum_{i=1}^{L}k_i\cdot 1_{\{t\in D_i\}},$$
$$\cdot Z(t)=(1_{\{t\in D_i\}},\dots 1_{\{t\in D_L\}},k=(k_1,\dots k_L)^T), \tag{6}$$

where k_i denotes the range of influence of holidays, $k\sim Normal(0,v^2)$, and v is affected by the $s(t)$ holidays_prior _scale metric.

We use the Seq2seq model to forecast the decomposed sequences. The encoder and decoder of Seq2seq are connected by an encoding vector, generally the last hidden state value h_t of the encoder, the transformation $q(h_t)$ of the previously hidden state value, or the transform of all hidden states.

The encoder and decoder use LSTM as the feature extractor. Each cell of the LSTM model has a containment gate structure and memory vector. The calculation process of the three-door structure is as follows (7).

$$\begin{aligned}\mathrm{f_t} &=\sigma(W_f\cdot[C_{t-1},h_{t-1},x_t]+b_f),\\ \mathrm{i_t} &=\sigma(W_i\cdot[C_{t-1},h_{t-1},x_t]+b_i),\\ \mathrm{f_o} &=\sigma(W_o\cdot[C_t,h_{t-1},x_t]+bo),\end{aligned} \tag{7}$$

where $\mathrm{f}_f, \mathrm{i}_i, \mathrm{f}_o$ represent the forgetting gate, the input gate, and the output gate, respectively. W_f, W_i, W_o are all matrices of dimension $[d, d+n]$, h_{t-1} is the history vector of the previous cell, and x_t is the input vector of the current cell. The output of the current cell is represented as (8).

$$\begin{aligned}\mathrm{z}_t &=\sigma(W_z\cdot[h_{t-1},x_t])\\ \widetilde{h}_t &=\tanh(W_z\cdot[h_{t-1}*r_t,x_t])\\ h_t &=h_{t-1}*(1-z_t)+\widetilde{h}_t*z_t\end{aligned} \tag{8}$$

The encoder receives all the input subsequences and outputs the corresponding coding vectors, and the decoder decodes the integrated coding vectors.

III. Experiment Preparation

A. Training and Test Samples

Generator power outages or temporary adjustments in the production environment of an offshore oil and gas field platform may result in abnormal data collection or data collection gaps. Therefore, the data must be preprocessed before model training. We use the 3σ principle to identify outliers and use the data from the time point before the missing values to replace the missing values. We then use the VMD and Prophet methods to decompose the preprocessed time-series data. Fig. 1 shows the sequence of intrinsic factors obtained after the decomposition of preprocessed time-series data. Intrinsic factor sequences cannot be used directly as input for model training. As shown in Fig. 2, we convert these sequences of intrinsic factors into supervised learning data. A fixed window intercepts the sequences into short sequences of fixed length, and the window keeps sliding to produce multiple consecutive short sequences. This transformation process is known as the sliding window algorithm. The input of the LSTM input contains a sample, time step, and feature. The data input process is shown in Fig.

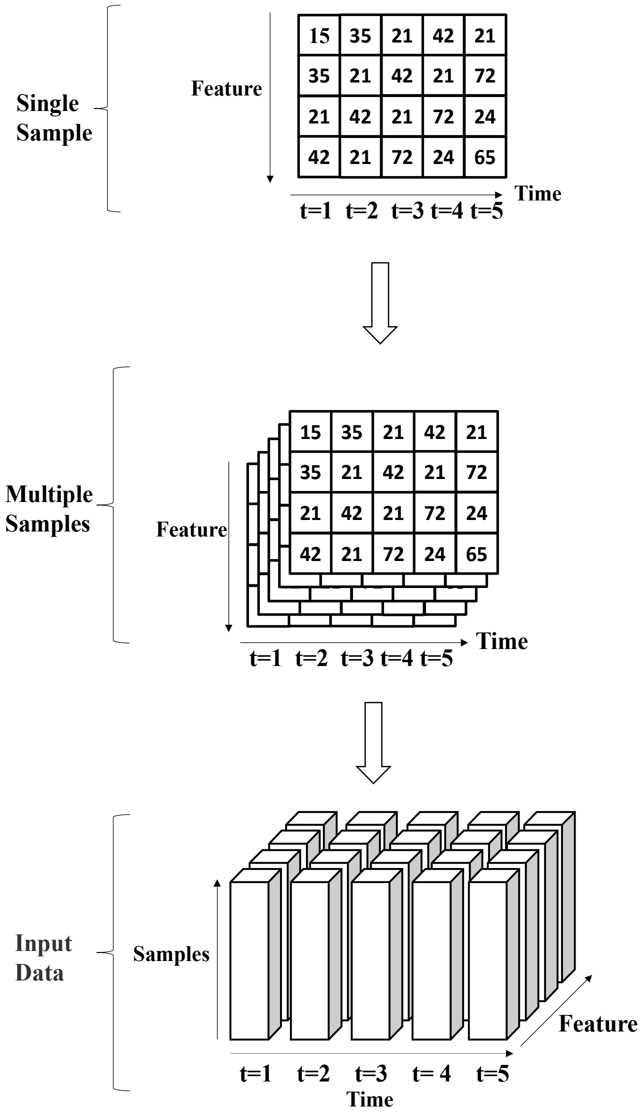


Fig. 3. Convert to 3D data.

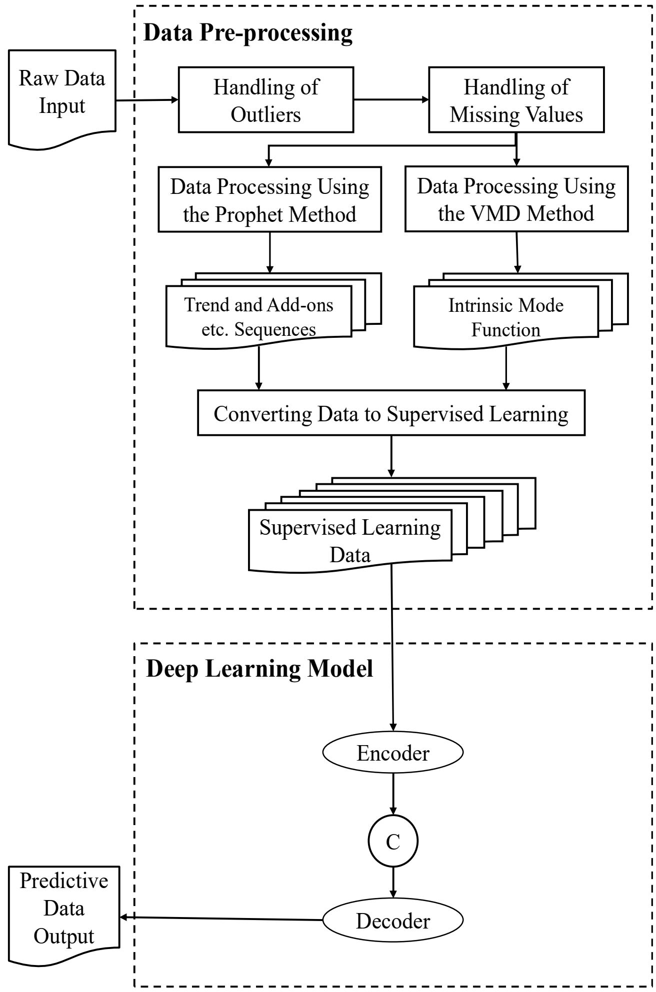


Fig. 4. Overall Forecast Scenario.

3. The processed dataset contains 104480 sample data points, of which 90% are training samples and 10% are test samples. We used the random crop method to crop the training and test samples to ensure a uniform distribution of the sample data.

B. Baseline

Recurrent neural networks are the first choice in time-series forecasting. Recurrent neural networks outperform ML and statistical methods in complex nonlinear time-series tasks [29],[30]. Hochreiter S et al. first proposed LSTM to improve RNNs [31]. Zheng et al. verified that LSTM has minor forecasting errors compared to ML models for complex power load data [32]. LSTM solves the gradient explosion and disappearance of RNNs. As a result, LSTM methods are able to remember longer time-series information. Zhang Yu et al. introduced the Seq2seq structure in short-term wind power forecasting [33]. The Seq2seq model uses LSTM as an encoder and decoder, and the authors showed that Seq2seq forecasts better than LSTM for longer sequences.

Time-series decomposition most often uses the signal and STL decomposition method. Signal decomposition methods decompose an original signal sequence into several more concise subsignal sequences. Niu et al. [34] used a combined VMD method and neural network model to forecast time-series data. The authors concluded that the combined model consisting of the VMD method and the neural network model had a smaller forecasting error than the single neural network model. The Prophet method is based on the STL method. The STL method uses robust locally weighted regression as the smoothing method.

From the above, we use the three models Seq2seq, VMD-Seq2seq, and Prophet-Seq2seq as the baseline.

C. Experiment Overall Design

The forecasting scheme in this paper includes a data preprocessing part based on data decomposition and a data training forecasting part.

Fig. 4 shows the forecasting process of the model. We detect and process outliers and missing values in the original time-series data. Then, we use the VMD method to decompose the preprocessed data into multiple IMFs and the Prophet method to decompose the preprocessed data into sequences such as trends and add-ons. Finally, we use the sliding window method to convert these decomposed sequences into supervised learning data.

The DL model is shown in Fig. 5. The encoder uses LSTM as the feature extractor. Each sub-LSTM in the encoder uses supervised learning data as input. The output of each sub-LSTM is the output of the last time step, and these outputs can be considered subcoding vectors. The output of the encoder is the sum of all subcoding vectors. The decoder uses this encoding vector as input for each time step. The decoder contains only one LSTM module, and the output of

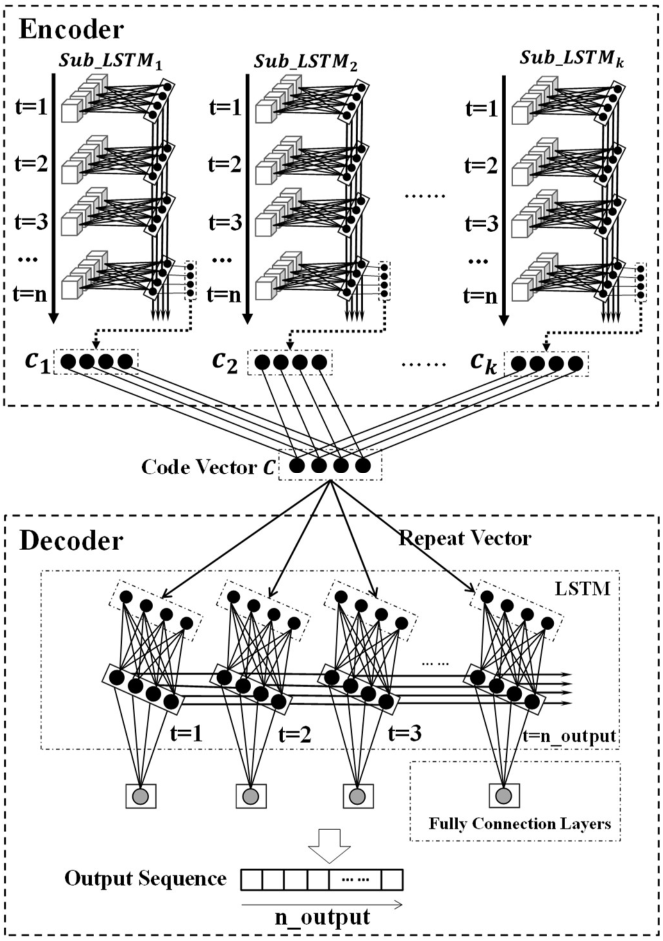


Fig. 5. Deep learning models used for forecasting.

each time step of LSTM corresponds to a fully connected layer. The final step of the decoder is to combine the forecast sequences based on the outputs of these fully connected layers.

IV. Experiments and Analysis

This section mainly involves the relevant empirical process and results. The experiment is conducted using a Windows 10 operating system with an Intel i7-8700K processor, 32 GB of RAM, and a GTX1070ti GPU.

A. Experimental Setup and Results

In this paper, we carried out two groups of controlled experiments. The first group of experiments compared the forecasting errors of the four models, Seq2seq, VMD-Seq2seq, Prophet-Seq2seq, and VMD-prophet-Seq2seq. The original power load data was processed into multiple sequences using VMD and Prophet methods, and the Seq2seq model was used to forecast the subsequences. In this set of experiments, the LSTM module had 60 units, and the number of IMFs in the VMD method was 6. The experimental results are shown in Fig. 6, Fig. 7, and Fig. 8. In addition, another set of experiments explored the effect of the number of LSTM units in the VMD-Prophet-Seq2seq model on the forecasting effect. The experiments set five sets of parameters with different units, and the numbers of units were 40, 50, 60, 70, and 80. The resulting graph is shown in Fig. 9. In this paper, three evaluation metrics, the root mean square error (RMSE), mean

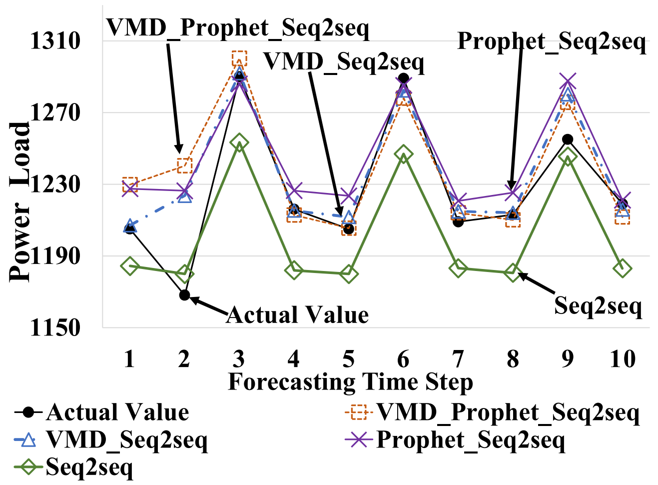


Fig. 6. Forecasting the predictive effect of each model when the sequence length is 10.

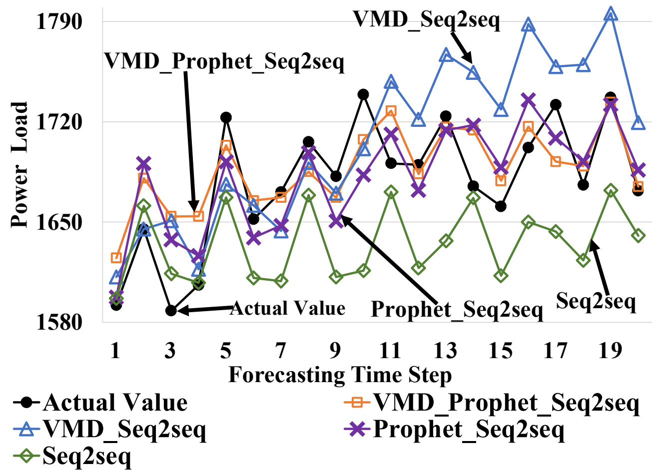


Fig. 7. Forecasting the predictive effect of each model when the sequence length is 20.

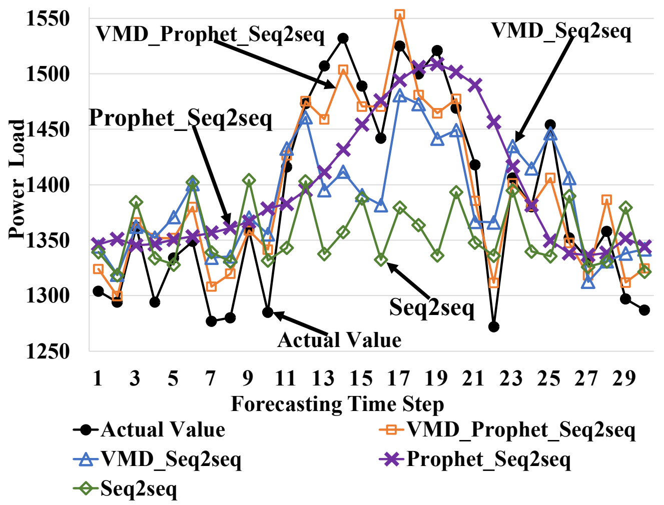


Fig. 8. Forecasting the predictive effect of each model when the sequence length is 30.

absolute error (MAE) and mean absolute percentage error (MAPE), were used.

The results of the first group of experiments are shown in Table I. When different forecasting steps were chosen, all three evaluation metrics of VMD-Prophet-Seq2seq were the smallest, and the Seq2seq model had the highest evaluation

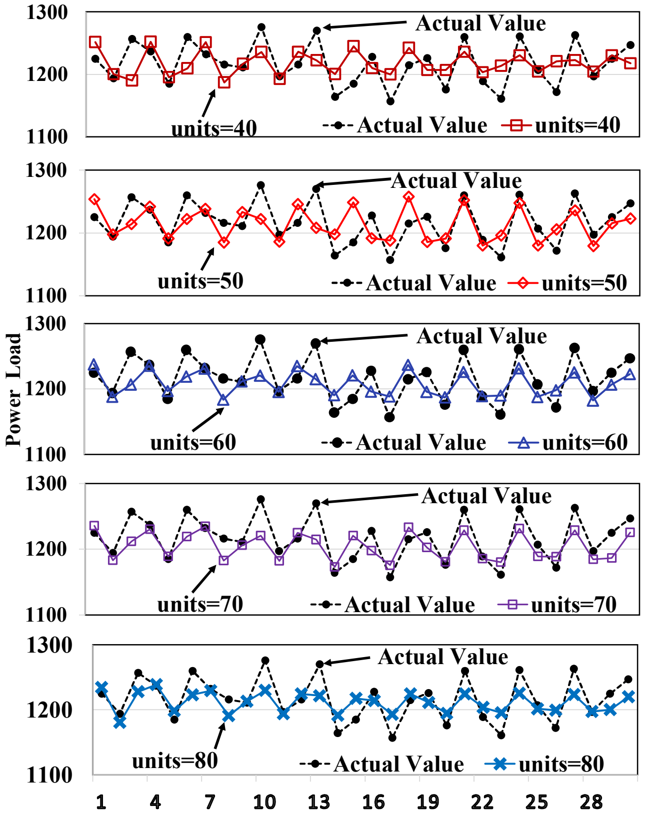


Fig. 9. Line graph of the prediction results of the VMD-Prophet-Seq2seq model using different numbers of neurons.

metrics among the four models. The evaluation metrics of the VMD-Prophet-Seq2seq model for forecasting time steps of n=10 and n=20 were not significantly different. A similar pattern of results was obtained in the VMD_Seq2seq model. We speculate that this might be due to a forecasting sequence that is too short. For the same evaluation indicator, the indicator values of all four models increase with the increase in the forecasting step.

Table II shows the results of the second group of experiments. We added multiple sets of comparison experiments containing different numbers of LSTM units. The results confirm that increasing the number of LSTM units decreases the forecasting error. We obtained good results with this simple method. Another new finding is that the model's forecasting error increases rather than decreases when the units slightly increase from 70 to 80.

B. *Model Principles and Analysis*

Validating these models in real scenarios is crucial for selecting the best model. Among the different decomposition methods described in Table 1, the VMD-Prophet-Seq2seq model has a minor forecasting error and, in combination with the experiments in Table 2, has the best results when the number of units is 70. Therefore, VMD-Prophet-Seq2seq is more suitable for offshore platform power load forecasting when the number of units is 70.

Based on the empirical findings above, the proposed research methodology can be applied to time-series data that is influenced by seasonal and holiday factors, such as merchandise sales volume. We use single-feature time-series data, which the model can use to forecast more accurately when more multidimensional features are not available. For a similar single-feature data scenario, given a single-feature

TABLE I
MODEL PREDICTION RESULT INDICATORS

Model \ Indices		RMSE	MAE	MAPE
Seq2seq	n=10	69.348	50.800	3.491%
	n=20	76.104	55.701	3.831%
	n=30	84.226	61.696	4.226%
VMD-Seq2seq	n=10	48.645	36.867	2.538%
	n=20	47.779	36.694	2.542%
	n=30	49.062	37.684	2.599%
Prophet-Seq2seq	n=10	49.287	37.304	2.580%
	n=20	53.084	40.572	2.800%
	n=30	57.207	44.888	3.098%
VMD-Prophet-Seq2seq	n=10	**45.153**	**34.898**	**2.411%**
	n=20	**45.494**	**34.920**	**2.414%**
	n=30	**46.192**	**35.474**	**2.450%**

TABLE II
MODEL REPRESENTATION OF DIFFERENT NUMBER OF UNITS

Units	RMSE	MAE	MAPE
40	51.260	39.108	2.682%
50	48.666	37.431	2.585%
60	46.192	35.474	2.450%
70	**45.087**	**34.581**	**2.389%**
80	45.137	34.697	2.393%

power load sequence $L_N:\{x_1,x_2,\ldots,x_N,\}$ of length N, k of units $IMF_k:\{x_{k,1},x_{k,2},\ldots,x_{k,N}\}$ of length N are obtained using VMD and Prophet decomposition. For each component, IMF_k is converted into a subset of l subsequences according to the sliding window method, as shown in Fig. 2, and each subsequence can be expressed as an $IMF_{l,k}:\{x_{k,sub-i},x_{k,sub-i+1},\ldots,x_{k,sub-1},x_{k,sub},\ldots,x_{k,sub+o}\}$, where i is the input length and o is the output length. The above method transforms the single-feature sequence into multiple subsets, and each subset can be represented as a single-feature data subset $D^{n\times i}$ with n samples and length i. Subset D is denoted $D=\{D_1,D_2,\ldots,D_k\}$.

In the encoder part of the Seq2seq model, each subset is taken as the input of a single LSTM unit, and each subset yields a context vector C_k. Then, the context vectors of each subset are cumulated to obtain a new context vector C, calculated as follows (9).

$$\begin{aligned} c_k &= LSTM\ (D_i) \\ \cdot C &= \sum_{i=1}^{k} c_k . \end{aligned} \tag{9}$$

In the decoder part, the coding vector C is taken as input for each time step of the LSTM unit in the decoder. The time step of the LSTM unit is equal to the number of forecasted time steps q. Each time step of the LSTM unit outputs the number of time steps through the fully connected layer of a single neuron, and these values are combined to form the output sequence $O=\{O_1,O_2,\ldots,O_q\}$.

As shown in Fig. 1, we obtained ten subsequences using the VMD and Prophet methods. Among the six subsequences decomposed by the VMD method, the IMF1 series exhibits nonperiodicity, and the five subsequences IMF2, IMF3, IMF4, and IMF5 exhibit some periodicity. Similarly, among the four subsequences decomposed by the Prophet method, additive_terms, daily and weekly show some periodicity, while trend does not show periodicity. The encoder uses LSTM as a feature extractor to more accurately capture the patterns of periodic data. The error of feature extraction for data with significant periodicity is significantly smaller than that for the original complex data. The two nonperiodic sequences of IMF1 and 'trend' obtained by decomposition are more compact than the original sequences. The extractor with the same parameters extracts the compact subsequence from the decomposition better than the original data extraction. Therefore, the feature extraction of the more compact subsequence can obtain more accurate feature information, and the accumulation of the feature information helps in obtaining smaller forecasting errors.

V. CONCLUSION

We proposed a combined VMD-Prophet-Seq2seq model that combines DL and decomposition methods. The model was suitable for multistep time-series forecasting scenarios with a single function. We verified that the combined model of the time-series decomposition method and deep learning model could improve the accuracy of time-series prediction by increasing the number of time-series decomposition methods used. Future research will focus on further feature engineering, using data dimensionality reduction to decrease the training time of the model.

REFERENCES

[1] L. Suganthi and A. A. Samuel, "Energy models for demand forecasting—A review," Renewable and Sustainable Energy Reviews, vol. 16, no. 2, pp1223-1240, 2012
[2] S. N. Hassan, M. H. Ahmad and N. Mohamed, "A comparison of the forecast performance of double seasonal ARIMA and double seasonal ARFIMA models of electricity load demand," Applied Mathematical Sciences, vol. 6, no. 135, pp6705-6712, 2012
[3] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," European Transport Research Review, vol. 7, no. 3, pp1-9, 2015
[4] B. Yildiz, J. I. Bilbao and A. B. Sproul, "A review and analysis of regression and machine learning models on commercial building electricity load forecasting," Renewable and Sustainable Energy Reviews, vol. 73, pp1104-1122, 2017
[5] M. Elena, M. H. Lee, H. Suhartono, I. Hossein, N. H. Abd Rahman and N. A. Bazilah, "Fuzzy time series and sarima model for forecasting tourist arrivals to bali," Jurnal Teknologi, vol. 57, no. 1, 2012
[6] A. Mosavi, P. Ozturk and K.-w. Chau, "Flood prediction using machine learning models: Literature review," Water, vol. 10, no. 11, pp1536, 2018
[7] P. Jiang, H. Yang and J. Heng, "A hybrid forecasting system based on fuzzy time series and multi-objective optimization for wind speed forecasting," Applied Energy, vol. 235, pp786-801, 2019
[8] L. Nguyen and V. Novák, "Forecasting seasonal time series based on fuzzy techniques," Fuzzy Sets and Systems, vol. 361, no. 1, pp114-129, 2019
[9] H. J. Sadaei, R. Enayatifar, F. G. Guimarães, M. Mahmud and Z. A. Alzamil, "Combining ARFIMA models and fuzzy time series for the forecast of long memory time series," Neurocomputing, vol. 175, pp782-796, 2016
[10] O. B. Sezer, M. U. Gudelek and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," Applied Soft Computing, vol. 90, pp106181, 2020
[11] J. Moon, J. Park, E. Hwang and S. Jun, "Forecasting power consumption for higher educational institutions based on machine learning," The Journal of Supercomputing, vol. 74, no. 8, pp3778-3800, 2018
[12] Z. Xiang, J. Yan and I. Demir, "A rainfall‐runoff model with LSTM‐based sequence‐to‐sequence learning," Water Resources Research, vol. 56, no. 1, ppe2019WR025326, 2020
[13] A. H. Bukhari, M. A. Z. Raja, M. Sulaiman, S. Islam, M. Shoaib and P. Kumam, "Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting," IEEE Access, vol. 8, pp71326-71338, 2020
[14] B. S. Freeman, G. Taylor, B. Gharabaghi and J. Thé, "Forecasting air quality time series using deep learning," Journal of the Air & Waste Management Association, vol. 68, no. 8, pp866-886, 2018
[15] T. Li, M. Hua and X. Wu, "A hybrid CNN-LSTM model for forecasting particulate matter (PM2. 5)," IEEE Access, vol. 8, no. 1, pp26933-26940, 2020
[16] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, vol. 454, no. 1971, pp903-995, 1998
[17] X. Qiu, Y. Ren, P. N. Suganthan and G. A. Amaratunga, "Empirical mode decomposition based ensemble deep learning for load demand time series forecasting," Applied Soft Computing, vol. 54, no. 1, pp246-255, 2017
[18] X. Kong, C. Li, F. Zheng, L. Yu and X. Ma, "Short-term load forecasting method based on empirical mode decomposition and feature correlation analysis," Automation of Electric Power Systems, vol. 43, no. 5, 2019
[19] Y. Wei and M.-C. Chen, "Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks," Transportation Research Part C: Emerging Technologies, vol. 21, no. 1, pp148-162, 2012
[20] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," IEEE Transactions on Signal Processing, vol. 62, no. 3, pp531-544, 2013
[21] S. A. Gyamerah, "On forecasting the intraday Bitcoin price using ensemble of variational mode decomposition and generalized additive model," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 3, pp1003-1009, 2020
[22] R. B. Cleveland, W. S. Cleveland, J. E. McRae and I. Terpenning, "STL: A seasonal-trend decomposition," J. Off. Stat, vol. 6, no. 1, pp3-73, 1990
[23] S. B. Taieb, G. Bontempi, A. F. Atiya and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition," Expert Systems with Applications, vol. 39, no. 8, pp7067-7083, 2012
[24] L. Cao, "Support vector machines experts for time series forecasting," Neurocomputing, vol. 51, pp321-339, 2003
[25] H. Hewamalage, C. Bergmeir and K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," International Journal of Forecasting, vol. 37, no. 1, pp388-427, 2021
[26] A. Rahman, V. Srikumar and A. D. Smith, "Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks," Applied Energy, vol. 212, no. 1, pp372-385, 2018
[27] F. Shahid, A. Zameer and M. Muneeb, "Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM," Chaos, Solitons & Fractals, vol. 140, no. 1, pp110212, 2020
[28] N. ur Rehman and H. Aftab, "Multivariate variational mode decomposition," IEEE Transactions on Signal Processing, vol. 67, no. 23, pp6039-6052, 2019
[29] M. Cai, M. Pipattanasomporn and S. Rahman, "Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques," Applied Energy, vol. 236, pp1078-1088, 2019
[30] M.-L. Shen, C.-F. Lee, H.-H. Liu, P.-Y. Chang and C.-H. Yang, "Effective multinational trade forecasting using LSTM recurrent neural network," Expert Systems with Applications, vol. 182, no. 1, pp115199, 2021
[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp1735-1780, 1997
[32] J. Zheng, C. Xu, Z. Zhang and X. Li, "Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network," in 2017 51st Annual Conference on Information Sciences and Systems (CISS), 2017. pp1-6
[33] Y. Zhang, Y. Li and G. Zhang, "Short-term wind power forecasting approach based on Seq2Seq model using NWP data," Energy, vol. 213, no. 1, pp118371, 2020
[34] H. Niu, K. Xu and W. Wang, "A hybrid stock price index forecasting model based on variational mode decomposition and LSTM network," Applied Intelligence, vol. 50, no. 12, pp4296-4309, 2020

Xingchen Song is a postgraduate student of Yangtze University. His current research interest is time-series forecasting.

Tongxi Wang received his B.S. and M.S. degrees from Chengdu University of Technology, Chengdu, China, in 1994 and 2007, respectively. He is currently an associate professor with the Department of Software Engineering, Yangtze University. He serves as the Head of the Department of Software Engineering, College of Computer Science and is an expert in doctoral and master's thesis review at the Degree Center of the Ministry of Education. He is also a member of the Chinese Association of Artificial Intelligence (CAAI). His current research interests are about big data, artificial intelligence technology and intelligent computing.

Ziming Xu is a postgraduate student of Yangtze University. His current research interest is computer vision, with specific interests in image processing, object detection and recognition.

Zhan Shi is a postgraduate student of Yangtze University. His current research interest is computer vision, with specific interests in image processing, object detection, recognition, and model compression.