Development and Comparison of Two Computational Intelligence Algorithms for Electrical Load Forecasts with Multiple Time Scales

Jinqi Liu
School of Engineering
University of Warwick
Coventry, UK
jinqi.liu@warwick.ac.uk

Sizhe Zhang
School of Engineering
University of Warwick
Coventry, UK
sizhe.zhang@warwick.ac.uk

Jihong Wang
School of Engineering
University of Warwick
Coventry, UK
jihong.wang@warwick.ac.uk

Abstract*— Electricity load forecasting provides the critical information required for power institutions and authorities to develop rational, effective, and economic dispatch plans. The load forecasting at the regional power system is important for optimal management and accommodating local renewable energy sources, which is a challenging task as the demand variations are more sensitive to local weather changes (such as temperature, humidity, precipitation, and wind speed) and consumers' activities and behaviours. The paper aims to develop a new prediction method using intelligent computational algorithms. Long Short-Term Memory (LSTM), a deep recurrent neural network, explores the long-term dependency of network memory sequence data to identify intrinsic variations in both horizontals (time series) and vertical (network depth) dimensions over a longer historical period. Support Vector Machine (SVM) is a typical learning method that has been successfully implemented to solve nonlinear regression and time series problems. This paper studies the two methods and adapts the two methods to become suitable algorithms for load prediction. The paper presents the algorithms, their applications and prediction results. The prediction performance is compared for using LSTM and SVM at ultra-short, short-term, medium-term, and long-term forecasting. The results show that LSTM has higher prediction accuracy than SVM in both ultra-short and short-term forecasts, but SVM is more capable of medium-term and long-term forecasting. Finally, the epoch time for LSTM and SVM is also calculated and compared.

***Keywords*—Load prediction; LSTM; SVM**

I. Introduction

Population increase, technology development, lifestyle changes and demand for various resources have imposed a significant impact on the environment, the most notable of which is Global Warming. Electricity generation has been considered as one of the primary causes of carbon emissions [1]. In May 2019, the UK Committee on Climate Change (CCC) amended the 2008 Climate Change Act and revealed the more ambitious goal of achieving net-zero carbon emissions by 2050 [2]. Power generation from Renewable Energy Sources (RESs) rapidly increases and imposes significant pressure on maintaining the grid stability as it is a great challenge of maintaining the balance between the load and generation due to the intermittent nature of RESs and the reduction of grid system inertia or spin reserve from the rotating machines [3]. Therefore, the cost of grid balance has continuously increased in recent years, which cost UK National Grid £1.789 billion in 2020 (49.3% year-on-year increase) [4]. The balance cost increased the electricity price and deprived the potential economic benefit of renewable energy.

A new operating model for modern grids or local power systems is needed. The prediction of load demand, electricity generation, and even market prices is essential for the management and operation of the future grid, determining the dispatch of the consumption and generation of electricity. The load demand prediction can be considered as forecasting using time series data. A time series is a series of discrete data with a continuous-time index at the same time intervals [5]. The data varies over time and there is some correlation between the data before and after the time scale. The analysis of time series has been widely used in areas such as signal processing, earthquake prediction, financial mathematics, weather forecasting and power system management [6]. The methodology for the prediction time series is shown in Fig. 1, which presents the most popular methods over the last two decades [7]. Artificial neural networks (ANN) ANN is a machine learning-based modelling approach that mimics the information is transmitted between neurons in the human brain. The neurons (also called activation function) at each layer are connected and have different connection weights (called synaptic weights). ANN is constantly updated to keep the network moving in the right direction by updating the connection weights through backpropagation [8]. Younes adopted generalised regression neural network (GRNN) to forecast plant disease by monitoring leaf wetness in 1999 [9]. Over the last two decades, deep learning has developed different layers and function blocks, which has led to its wide use in time series prediction [10]. For instance, Takashi used a Deep Trust Network (DBN) consisting of a stack of Restricted Boltzmann Machines (RBMs) to make

Identify applicable funding agency here. If none, delete this text box.

predictions on time series [11]. Long Short-term Memory (LSTM) is designed to address the need for the model to actively select helpful information from historical data by using different gates [12].

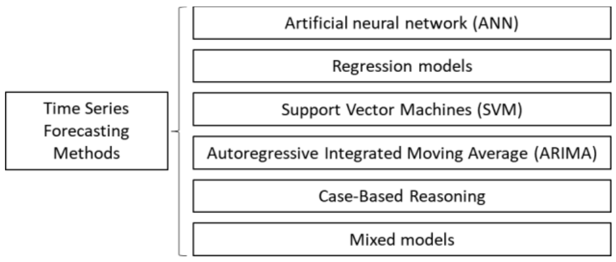
Time Series Forecasting Methods
Artificial neural network (ANN)
Regression models
Support Vector Machines (SVM)
Autoregressive Integrated Moving Average (ARIMA)
Case-Based Reasoning
Mixed models

Fig. 1. The methodology for prediction time series.

The Support Vector Machine (SVM) was developed based on the concept of a decision hyperplane that divides the data into different groups by Vapnik and Alexey in 1963 [13]. The hyperplane is defined by finding the maximum margin (distance) between the different groups. SVMs can also fit the parameters of linear and nonlinear regression models by mapping the inputs to a high-dimensional feature space. Kyoung-jae uses SVM to predict the stock price index and finds the performance of SVM is better than three-layer backpropagation networks at 5% statistical significance level [14]. Hong developed an SVM regression model that uses an immune algorithm (IA) to determine model parameters to predict the annual electricity load in Taiwan [15]. Deng combined particle swarm optimisation (PSO) and square support vector machine (LS-SVM) to estimate the fault of rotating machinery with the highest 89.5% of accuracy rate among 11 compared algorisms [16].

In this paper, the development of load prediction algorithms uses the University of Warwick campus energy data as a representative case in algorithms refinement and verification.

II. DEVELOPMENT OF COMPUTATIONAL INTELLIGENCE ALGORITHMS FOR LOAD FORECASTING

A. *The principle of LSTM*

Recurrent Neural Network (RNN) structure mimics the thought pattern that human cognition is based on experience and memory, and is mainly used to process contextually relevant sequential data [17]. In an RNN, it is not only the input information at this time that is considered but also the previous input information that is retained through the function of "memory". The outputs corresponding to the network's historical input data are weighted and fed back into the training network at this moment. The series-connected network structure is suitable for processing data with contextual information. The number of training parameters and training time can be reduced as the parameters of RNN can be shared between the same layers. The RNN has been widely used in fields such as natural language processing and speech recognition due to these advantages in recent years [18]. A basic RNN network structure mainly composed of an input layer, hidden layer, and output layer.

The RNN can be expressed as:

$$y_t = f_y(a_k * S_t) \tag{1}$$

$$S_t = \sum_{k=1}^{n} f_k(a_k(i * x_t + w_k * s_k) + b_k) \tag{2}$$

where the i, a_k, and b_k are the weight and bias between each layer, while w_k is the weigh within each layer. x_t and y_t are the input and output of the network, respectively. n represents the layer number of hidden layers, and $f_{z(z=x,y,s_k)}$ indicates the activation function at each layer. Though the intra-layer weights w_k, the function of "remembering" the previous input is achieved. The difference between RNN and FNN it can be observed: RNN adds additional inter-layer weights and returns the historical results at the hidden layer to the current hidden layer to "remember" the historical data. Substituting (2) into (1):

$$y_t = f_y\left(a_k * \sum_{k=1}^{n} f_k(a_k(i * x_t + w_k * s_k) + b_k)\right) \tag{3}$$

As seen from (3), the output of RNN at the time t is determined by both the previous input $(x_{t-1}, x_{t-2}, \ldots)$ and the current input x_t. This is the reason why RNN is able to remember previous input and learn the correlation between the data time series.

RNNs can theoretically solve long-term dependency problems: the gap between the relevant information and the point where it is needed is very large/far. However, in practice, it is worth noting that the original RNN model suffers from the pain of gradient vanishing or gradient exploding [19]. To solve the long-term memory problem, Hochreiter and Schmidhuber proposed the LSTM network [20]. LSTM network, as an improved RNN network, introduces the idea of self-looping, which generates paths that allow gradients to flow over long-time scales. The weights used to control the state of this loop and the duration of memory accumulation can change dynamically in the model, and the change depends on the contextual information of the data sequence. In terms of computation, the LSTM does not do multiplication to generate memory. Instead, the current input is superimposed with the results of past operations. This manner avoids the gradient vanishing or exploding. LSTM has achieved excellent performance in several fields at present, such as speech recognition, machine translation, and handwriting recognition. The structure of LSTM network is shown in Fig. 2. As can be seen from Fig. 2, the LSTM network contains forget gate G_t^f, input gate G_t^i, output gate G_t^o, and candidate states $\tilde{c_t}$. The input of the network contains the external input of the current moment and the output and cell state of the previous moment.

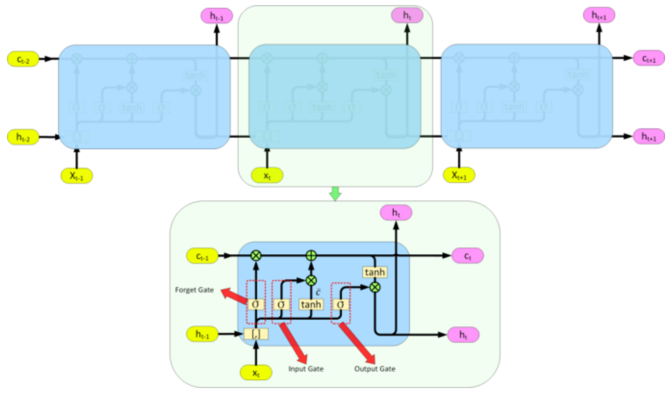

Fig. 2. LSTM network structure.

The equation of each gate is:

$$G_t^f = \sigma\left(a_f \bullet [h_{t-1}, x_t] + b_f\right) \tag{4}$$

$$G_t^i = \sigma(a_i \bullet [h_{t-1}, x_t] + b_i) \tag{5}$$

$$G_t^o = \sigma(a_o \bullet [h_{t-1}, x_t] + b_o) \tag{6}$$

where h_{t-1} and x_t are the output of last state and present input, respectively. The $a_{k(k=f,i,o)}$ and $b_{k(k=f,i,o)}$ are the weight and bias for each gate. The candidate states $\widetilde{c_t}$, cell states c_t, and output h_t can be written as:

$$\widetilde{c_t} = tanh(a_c \bullet [h_{t-1}, x_t] + b_c) \tag{7}$$

$$c_t = G_t^f * c_{t-1} + G_t^i * \widetilde{c_t} \tag{8}$$

$$h_t = G_t^o * tanh(c_t) \tag{9}$$

With the weight a and bias b, and activation functions σ for each gate, the optimisation of LSTM models can be conducted by minimising the cost function (CF). The output of ANN can be expressed as:

$$\tilde{y} = f(a_{k(k=f,i\ o)}, b_{k(k=f,i\ o)}, \sigma_{k(k=f,i\ o)}) \tag{10}$$

B. The principle of SVM

SVM methods can be divided into two main types, Support Vector Classification (SVC) and Support Vector Regression (SVR). SVR is used to perform the load prediction on multiple time scales in this paper.

Suppose that the training sample at moment n is:

$$z_n = \{(x_1, y_1), (x_2, y_2) \cdots (x_n, y_n)\} \tag{11}$$

where $x_n,\ i = 1, \dots \dots n$ are input samples and $y_n,\ i = 1, \dots \dots n$ are output samples. The core principle of both linear support vector regression and nonlinear regression is to obtain a smoothed regression function $f(x)$ by training the input samples so that the regression function given by $f(x)$ can minimise the error between the actual output samples and the predicted output samples.

Assume that the expression of the regression function for the nonlinear load prediction model is:

$$f(x) = \langle w \cdot \varphi(x_i) \rangle + b \tag{12}$$

where w is the weighting vector, b is the bias parameter, the $\langle \cdot \rangle$ denotes the inner product in feature space and the $\varphi(x_i)$ is the mapping from x_i to a high-dimensional linear Hilbert space [21]. To smooth the function $f(x)$, a minimum w needs to be found. One way to achieve this is to minimise the norm. Therefore, the optimisation problem of (12) can be transformed into the following form:

$$\min \frac{1}{2} \|w\|^2 \tag{13}$$

Subject to:

$$\begin{cases} y_i - \langle w \cdot \varphi(x_i) \rangle - b \le \varepsilon \\ \langle w \cdot \varphi(x_i) \rangle + b - y_i \le \varepsilon \end{cases} \tag{14}$$

In which ε is the insensitive loss function. The ε-region is enclosed by two parallel lines, the error of sample points located in the ε-region will be ignored. In other words, ε-region is a region that will not contribute any loss to the loss function. This concept is the insensitivity theory introduced by Vapnik [22].

Two slack variables ξ and ξ^* are introduced in (13) and (14) to ensure that the above optimisation problem has a solution:

$$\min[\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*)] \tag{15}$$

Subject to:

$$\begin{cases} y_i - \langle w \cdot \varphi(x_i) \rangle - b \le \varepsilon + \xi_i \\ \langle w \cdot \varphi(x_i) \rangle + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \tag{16}$$

where C is regularisation parameter, representing the degree of punishment for misclassified sample points [21]. Lagrange function can be taken to solve above problem:

$$L = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) - \sum_{i=1}^n \alpha_i (\varepsilon + \xi_i - y_i + \langle w \cdot \varphi(x_i) \rangle + b) - \sum_{i=1}^n \alpha_i^* (\varepsilon + \xi_i - y_i - \langle w \cdot \varphi(x_i) \rangle - b) - \sum_{i=1}^n (\eta_i \xi_i + \eta_i^* \xi_i^*) \tag{17}$$

where α_i, α_i^* and η_i, η_i^* are Lagrange multipliers. According to KKT condition, the following equations can be achieved [23]:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^n (\alpha_i - \alpha_i^*) \varphi(x_i) = 0 \tag{18}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0 \tag{19}$$

$$\frac{\partial L}{\partial \xi^{(*)}} = C - {\alpha_i}^{(*)} - {\eta_i}^{(*)} = 0 \tag{20}$$

Substituting (18), (19) and (20) into (17) can transform it into a dual optimisation problem:

$$maximize\ L = -\frac{1}{2} \sum_{i,j=1}^n (\alpha_i - \alpha_i^*) \left(\alpha_j - \alpha_j^*\right) K\left(x_i, x_j\right) - \varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*) \tag{21}$$

Subject to:

$$\begin{cases} \sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0 \\ \alpha_i - \alpha_i^* \in [0, C] \end{cases} \tag{22}$$

where the $K\left(x_i, x_j\right) = \langle \varphi(x_i) \cdot \varphi(x_j) \rangle$ is introduced in the nonlinear regression case to replace the complex inner product operation between the input samples in the original nonlinear space. The equation of w and the nonlinear load forecasting model can be expressed [21]:

$$\begin{cases} w = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \varphi(x_i) \\ f(x) = \sum_{i,j=1}^n (\alpha_i - \alpha_i^*) K(x_i \cdot x_j) + b \end{cases} \tag{23}$$

Bias parameter b can be calculated:

$$b = average\left|\varepsilon sgn(\alpha_i - \alpha_i^*) + y_i - \sum_{i,j=1}^n (\alpha_i - \alpha_i^*) K(x_i \cdot x_j)\right| \tag{24}$$

In the calculation of $f(x)$, there is no need to explicitly get the mapping function $\varphi(x)$. Only kernel function $K(x_i \cdot x_j)$ needs to be used, which simplifies the process.

The selection of kernel functions is crucial, there are many kernel functions that can be applied in different scenarios. The Gaussian Radial Basis Function (RBF) kernel is used in this paper:

$$K\left(x_i \cdot x_j\right) = \exp\left(-\left\|x_i - x_j\right\|^2 / 2\sigma^2\right) \tag{25}$$

In the load forecasting model of SVM, the regularisation parameter C and the width coefficient of RBF σ play essential roles in model's prediction performance. Therefore, to get to the optimal prediction accuracy, selecting optimal parameters C and σ is vital. The cross-validation method is addressed to determine C and σ.

III. THE PROCESS OF APPLYING TWO ALGORITHMS FOR LOAD FORECASTING

A. Data acquisition and feature extraction

The half-hourly recorded load data for 2020 and the first nine months of 2021 are used, which are collected from the

University of Warwick (UoW) campus energy consumption database. The data of 2020 performs as the training dataset and the data of 2021 as the test dataset.

The previous studies indicate that three sets of information are required to achieve more reliable prediction results: Historical load demands, Calendar information and Weather data [24]. Regarding the calendar information, four features are considered: the day label using an integer from 1 to 365 to mean 365 days through a year, week label with an integer from 0 to 6 to represent Sunday to Sunday, timeslot label in a day using integer 0 – 47 to denote 24 hours with 30 minutes time intervals, and three binary digits are taken to distinguish various types of days, such as the weekdays, weekends, term time and public holidays. In addition, temperature, humidity, and wind speed data are chosen as the representation of weather information.

B. Data processing

Data normalisation is to improve the learning model's numerical stability and speed up the training process. In this study, the sampled data are mapped to the [0,1] interval after the normalisation with the following equation:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (26)$$

where x_{max} represents the maximum value in the sample data and x_{min} denotes the minimum value in the sample data. x and x_{scaled} are values before and after the normalisation. Moreover, the data requires to be inversely normalised after the prediction according to the following equation:

$$x = x_{min} + x_{scaled}(x_{max} - x_{min}) \quad (27)$$

Furthermore, a few outliers exist in the data obtained due to the mismeasurement of utilities on the campus. The outliers require to be removed since their interference with the training process and prediction accuracy. The removal of outliers is performed through the linear fitting of one load value before and after the outliers.

C. Error evaluation index

To evaluate the prediction accuracy of models, the error indexes Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) are used, the formula of them are shown below:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|A_i - P_i| \quad (28)$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|A_i - P_i|}{A_i} \times 100 \quad (29)$$

where n is the number of data, A_i means actual load values for i times, and P_i represents predicted load values for i times.

D. Prediction steps of two algorisms

The flow chart for LSTM and SVM prediction is shown in Fig. 3 and Fig. 4. According to the steps of SVM load prediction, after data collection and processing, training sample arrays can be achieved based on the actual load data and selected features. Then, after the determination of initial parameters and coefficients, an initial model for load prediction can be obtained. The optimisation is carried out based on overall prediction performance at last. As for the LSTM, hyperparameter has been set at first. The bias and weight are then randomly generated. The gradient descent is used to minimise the error between the actual value and prediction value. Error is a function of the weight and bias. In the proposed LSTM method, the Mini-batch gradient descent (MBGD) is adopted [25].

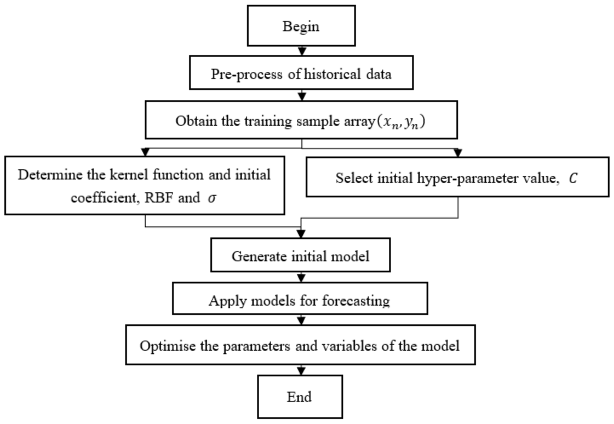
Begin
Pre-process of historical data
Obtain the training sample array (x_n, y_n)
Determine the kernel function and initial coefficient, RBF and σ
Select initial hyper-parameter value, C
Generate initial model
Apply models for forecasting
Optimise the parameters and variables of the model
End

Fig. 3. SVM prediction flow chart.

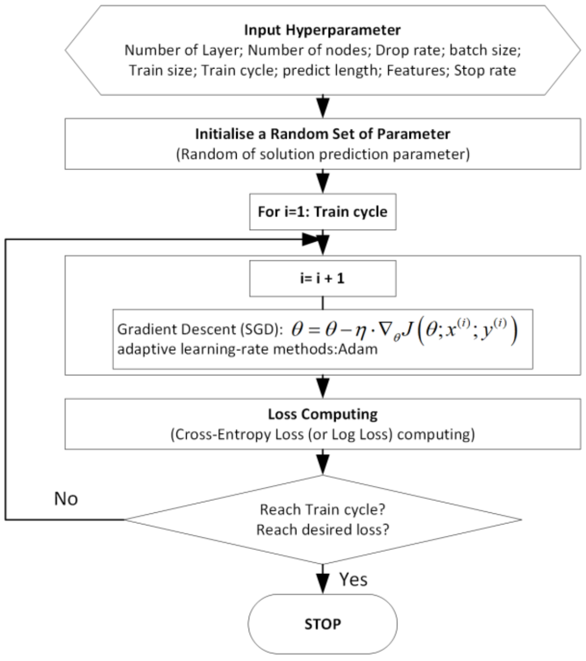
Input Hyperparameter
Number of Layer; Number of nodes; Drop rate; batch size; Train size; Train cycle; predict length; Features; Stop rate
Initialise a Random Set of Parameter
(Random of solution prediction parameter)
For i=1: Train cycle
i= i + 1
Gradient Descent (SGD): $\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i)}; y^{(i)})$
adaptive learning-rate methods:Adam
Loss Computing
(Cross-Entropy Loss (or Log Loss) computing)
No
Reach Train cycle?
Reach desired loss?
Yes
STOP

Fig. 4. LSTM prediction flow chart.

IV. Presentation and Comparison of Prediction Results

Four different prediction time horizon scenarios are selected based on various prediction applications:

- Ultra-short-term load forecasting (USTLF): Predict load value after half an hour
- Short-term load forecasting (STLF): Predict load values in the next day
- Medium-term load forecasting (MTLF): Predict load values in the next week

- Long-term load forecasting (LTLF): Predict load values in the next month

Four prediction modes can be applied to different scenarios. For instance, USTLF and STLF can help energy management manage electricity consumption and distribution in real-time. MTLF and LTLF can help control the planning of electricity consumption and generation for the energy system. Note that all prediction is conducted based on half-hour resolution, and the simulation in this paper is implemented in Python software.

One typical day is chosen for USTLF and STLF results, and a typical week and month are selected for the MTLF and LTLF. Four subplots in Fig. 5 and Fig. 6 from top to bottom correspond to USTLF, STLF, MTLF, and LTLF. The simulation in this paper is implemented in Python software. For the LSTM model in USTLF, data from the last 24 hours will be considered to forecast electricity demand for the next half hour, while for the SVM model, data from the past 48 hours are used. For both prediction methods, the STLF uses data from the past week to forecast, while the MTLF and LTLF use the past week and month data to predict. It can be found from two figures that the discrepancy between two curves turns greater with the increase of the prediction time scale, meaning a worse prediction accuracy in a longer prediction time horizon. It can be concluded that these two methods, LSTM and SVM, perform better for shorter time horizon load prediction. To compare the prediction performance of two methods, the training/epoch time and the model structure or hyper-parameters are shown in TABLE I.

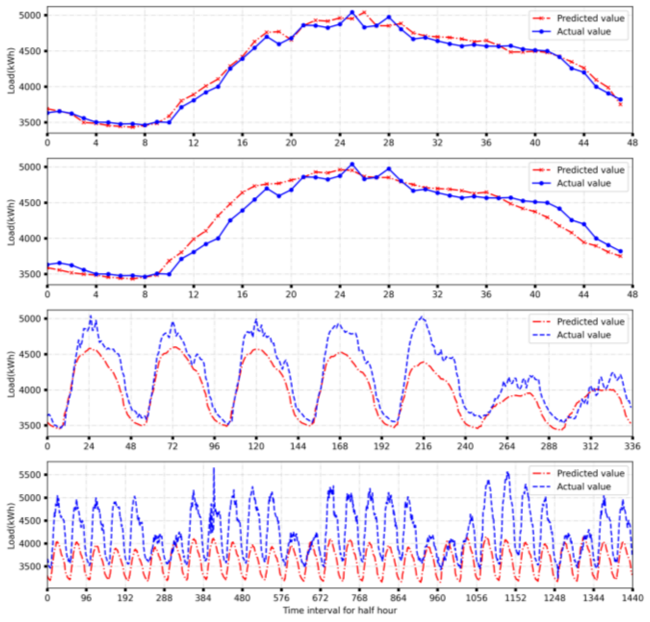

Fig. 5. Prediction results for four scenarios of LSTM.

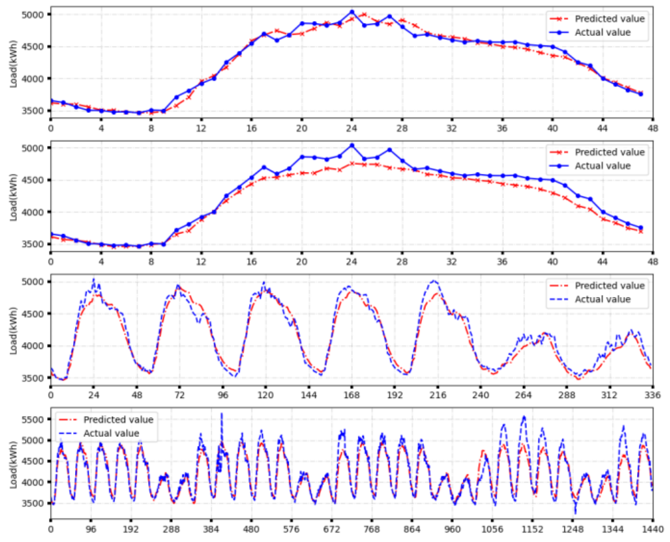

Fig. 6. Prediction results for four scenarios of SVM.

TABLE I. MODEL PARAMETERS AND TRAINING/EPOCH TIME FOR LSTM AND SVM.

LSTM				
	Layer number	Nodes number	Batch size	Epoch time (s)
USTLF	10	10	39	132
STLF	10	40	13	403
MTLF	10	50	6	494
LTLF	10	100	4	2085

SVM			
	C	σ	Epoch time (s)
USTLF	1	0.03	43
STLF	1	0.01	499
MTLF	0.2	0.01	631
LTLF	0.1	0.005	426

In addition, typical days, weeks, and months in four seasons are selected to verify the effectiveness of load prediction under various cases. The model metric (MAE and MAPE (%)) in four scenarios and the total test dataset are presented in TABLE II and Fig. 7. The epoch time is calculated on Intel® Xeon Silver 4114 CPU 2.20GHz and 64GB RAM with Windows Server 2019 system.

TABLE II. MODEL METRIC FOR LSTM AND SVM IN VARIOUS SCENARIOS

LSTM						
		Spring	Summer	Autumn	Winter	Test dataset
USTLF	MAE	0.014	0.012	0.08	0.011	0.010
	MAPE	3.507	2.997	2.001	2.752	2.501
STLF	MAE	0.011	0.015	0.012	0.019	0.014
	MAPE	2.761	3.751	3.002	4.752	3.477
MTLF	MAE	0.074	0.122	0.142	0.135	0.110
	MAPE	18.507	30.512	35.512	33.763	27.595
LTLF	MAE	0.279	0.342	0.266	0.321	0.307
	MAPE	69.778	85.537	66.526	80.28	73.722
SVM						
		Spring	Summer	Autumn	Winter	Test dataset

USTLF	MAE	0.011	0.016	0.010	0.017	0.012
	MAPE	2.605	3.048	2.465	3.165	3.014
STLF	MAE	0.028	0.053	0.017	0.053	0.024
	MAPE	5.817	8.547	3.932	8.551	6.023
MTLF	MAE	0.023	0.043	0.044	0.044	0.037
	MAPE	6.079	8.985	11.877	10.766	9.308
LTLF	MAE	0.032	0.047	0.063	0.057	0.048
	MAPE	9.073	9.141	15.757	14.968	12.684

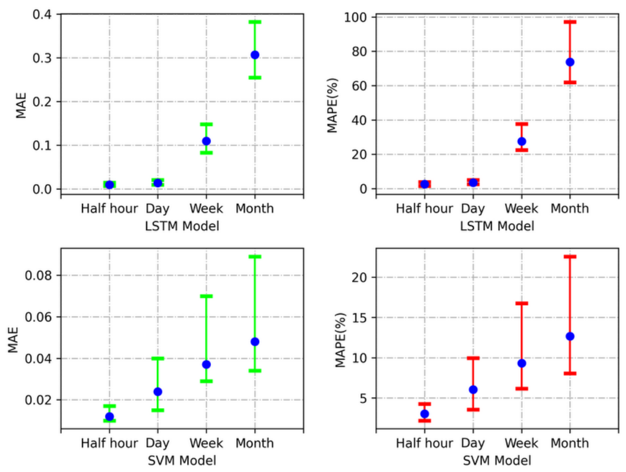

Fig. 7. MAE and MAPE value of prediction results and error range for four scenarios of LSTM and SVM.

As shown in TABLE II and Fig. 8, the predicted accuracy for LSTM for USTLF and STLF is lower than SVM, but in terms of the MTLF and LTLF, the SVM achieves a better result. The high accuracy of LSTM can be attributed to LSTM investigating the hidden information in the time series, and the deeper network levels can fit more complex nonlinear relationships. The lower accuracy for LSTM in MTLF and LTLF is due to the length of input data. Using a week or a month of data to predict may not be capable of capturing the features of the data. However, due to the better generalisation capability and the use of kernel function, the SVM method can jump out the local minimal and achieve the global optimal solution, making the SVM algorism more suitable for dealing with the data with the larger size, which may be the reason accounting for the better prediction performance for SVM in MTLF and LTLF. A more acceptable way for LSTM is to enlarge the length of input data. In accordance with the input data size, a more complex (more layers and node numbers) network should be established to understand the data. To train a more complex network, more computing power may be required. As shown in TABLE I, the training time for each epoch needs 2085 seconds with an input data length of one month. Complex models may require more computing power, and existing platforms' performance limits the exploration of more complex LSTM models. This also reveals the dependence of deep learning networks on computer computing power. LSTM models with more complex models and longer time scale inputs may be able to improve accuracy significantly. However, SVM does have a higher performance than LSTM in MTLF and LTLF for the exact implementation platform.

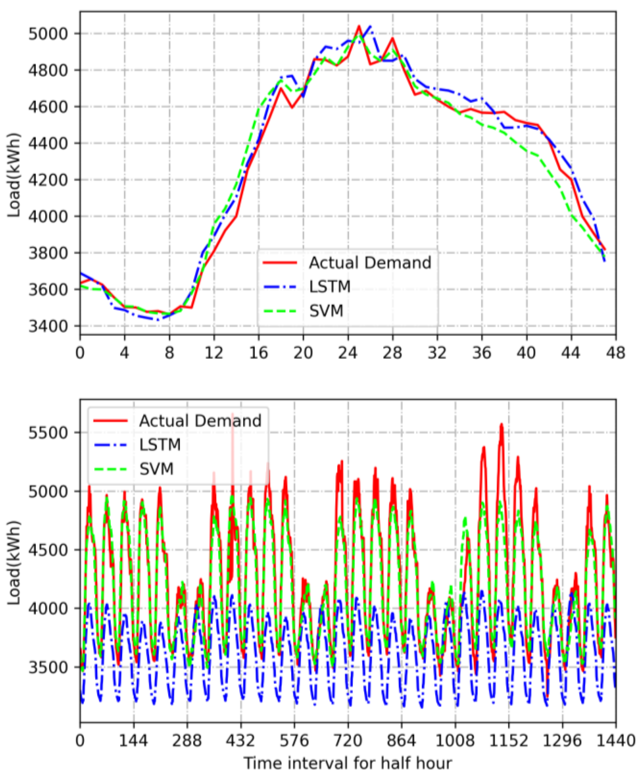

Fig. 8. Comparison of prediction results between LSTM and SVM for ultra-short-term and long-term load prediction.

V. Conclusion

In this paper, the principle of two prediction methods, LSTM and SVM, are selected and introduced. They are then formulated and modified for the applications of electrical load prediction in four different time scales with selected features extracted from the actual data. The studies found that both methods are suitable for shorter time horizon prediction. Their prediction performance, including the prediction accuracy and training/epoch time, indicates that the LSTM has a higher accuracy for ultra-short- and short-term load prediction, while the SVM method performs better for medium- and long-term prediction. Meanwhile, the epoch time for the SVM method is shorter than LSTM in most cases, which is particularly evident in the case of medium and long-term prediction. The study has also showed that the prediction accuracy of LSTM is highly relevant to the computing power of the computer.

References

[1] R. Carmichael, "Behaviour change, public engagement and Net Zero, a report for the Committee on Climate Change," 2019.

[2] F. Tanneberger *et al.*, "Towards net zero CO2 in 2050: An emission reduction pathway for organic soils in germany," *Mires and Peat,* vol. 27, 2021.

[3] X. Qin *et al.*, "Study on inertia support capability and its impact in large scale power grid with increasing penetration of renewable energy sources," in *2018 International Conference on Power System Technology (POWERCON)*, 2018: IEEE, pp. 1018-1024.

[4] M. Nedd *et al.*, "Operating a zero-carbon GB power system: implications for Scotland," University of Strathclyde, 2020.

[5] C. Chatfield, *Time-series forecasting*. CRC press, 2000.

[6] Z. Hajirahimi and M. Khashei, "Hybrid structures in time series modeling and forecasting: A review," *Engineering Applications of Artificial Intelligence,* vol. 86, pp. 83-106, 2019.

[7] H. Liu, G. Yan, Z. Duan, and C. Chen, "Intelligent modeling strategies for forecasting air quality time series: A review," *Applied Soft Computing,* p. 106957, 2021.

[8] N. Hecht, "Theory of the backpropagation neural network," in *International 1989 Joint Conference on Neural Networks*, 1989 1989, pp. 593-605 vol.1, doi: 10.1109/IJCNN.1989.118638.

[9] Y. Chtioui, S. Panigrahi, and L. Francl, "A generalized regression neural network and its application for leaf wetness prediction to forecast plant disease," *Chemometrics and Intelligent Laboratory Systems*, vol. 48, no. 1, pp. 47-58, 1999.

[10] Z. Han, J. Zhao, H. Leung, K. F. Ma, and W. Wang, "A review of deep learning models for time series prediction," *IEEE Sensors Journal,* 2019.

[11] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using a deep belief network with restricted Boltzmann machines," *Neurocomputing,* vol. 137, pp. 47-56, 2014.

[12] N. Somu, G. R. MR, and K. Ramamritham, "A hybrid model for building energy consumption forecasting using long short term memory networks," *Applied Energy,* vol. 261, p. 114131, 2020.

[13] B. Schölkopf, Z. Luo, and V. Vovk, *Empirical inference: Festschrift in honor of Vladimir N. Vapnik*. Springer Science & Business Media, 2013.

[14] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing,* vol. 55, no. 1-2, pp. 307-319, 2003.

[15] W.-C. Hong, "Electric load forecasting by support vector model," *Applied Mathematical Modelling,* vol. 33, no. 5, pp. 2444-2454, 2009.

[16] W. Deng, R. Yao, H. Zhao, X. Yang, and G. Li, "A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm," *Soft Computing,* vol. 23, no. 7, pp. 2445-2462, 2019.

[17] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235-1270, 2019.

[18] S. Karita *et al.*, "A comparative study on transformer vs rnn in speech applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019: IEEE, pp. 449-456.

[19] A. Kag, Z. Zhang, and V. Saligrama, "Rnns incrementally evolving on an equilibrium manifold: A panacea for vanishing and exploding gradients?," in *International Conference on Learning Representations*, 2019.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[21] X. Liao, X. Kang, M. Li, and N. Cao, "Short term load forecasting and early warning of charging station based on pso-svm," in *2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, 2019: IEEE, pp. 305-308.

[22] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 1999.

[23] S. Qiang and Y. Pu, "Short-term power load forecasting based on support vector machine and particle swarm optimization," *Journal of Algorithms & Computational Technology,* vol. 13, p. 1748301818797061, 2018.

[24] Y. Chen and D. Zhang, "Theory-guided deep-learning for electrical load forecasting (TgDLF) via ensemble long short-term memory," *Advances in Applied Energy,* vol. 1, p. 100004, 2021.

[25] D. Peng, T. Gu, X. Hu, and C. Liu, "Addressing the multi-label imbalance for neural networks: An approach based on stratified mini-batches," *Neurocomputing,* vol. 435, pp. 91-102, 2021.