2021 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT) | 978-1-6654-4950-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICMLANT53170.2021.9690555

Reinforcement learning for optimization hyperparameters of Long Short-Term Memory applied to Electricity load forecasting

Nguyen Thi Ngoc Anh
School of Applied Mathematics and Informatics
Hanoi University of Science and Technology
Hanoi, Vietnam
anh.nguyenthingoc@hust.edu.vn

Dang Tien Dat
School of Applied Mathematics and Informatics
Hanoi University of Science and Technology
Hanoi, Vietnam
shawcicdd@gmail.com

Le Anh Ngoc
Swinburne Vietnam
FPT University
Hanoi, Vietnam
ngocla2@fe.edu.vn

***Abstract*—Accurate electricity load forecasting is a key factor in optimizing the operation of the power system. Moreover, more accurate forecasting helps the better efficiency, reliability, and security of the power system. Uncertain weather and uncertain behavior of consumers is the reason for the uncertain behavior of electricity load. The accuracy forecast model is necessary to research. A new model for short-term electricity load forecast having better accuracy performance is investigated in this paper. The proposed method combines long-short term memory and reinforcement learning for short-term electricity load. Concretely, reinforcement learning is used for the optimization of hyperparameters of Long Short-Term Memory (LSTM). The proposed model is applied to the electricity load forecasting of Vietnam. The experimental results show that the proposed model has better performance than the pure LSTM model.**

***Keywords*—Electricity load, LSTM, reinforcement learning, hyperparameters, optimization**

I. Introduction

Electricity load forecasting is an important problem in the electricity system of the nation. Indeed, highly accurate short-term electricity load forecasting is necessary in the operation plan to produce and detect overload cases so the company can establish an operational plan as well as adjustment of load reduction and management of electricity prices. Currently, researchers use three main methods: i) Classical statistical analysis models such as AR, ARMA, ARIMA, ARIMAX, SARIMA [1], [2], [3]; (ii) Machine learning models such as ANN, RNN, LSTM, N-BEAT [4], [5], [2]; (iii) The hybrid model combines the strengths and weaknesses of classical statistical models and machine learning.

Nowadays, because of the development of technology, researchers need to build a model with higher accuracy by analyzing big data of smart devices such as electricity load data compiled in minutes, hours of many electric power plants and weather data, as well as events in social life. The problem requires choosing a suitable model with electricity load data which is collected as time series data.

Some classical models that have been studied for short-term load forecasting are AR, ARMA, ARIMA, SARIMA, ARIMAX. Classical time series models only use one series of data and assume the dataset is the set of observations of one time series variable consisting of three components: trend, seasonal, and a random component. Classic models assume a time series variable follows that model, and it is represented as autoregressive equations. The traditional model ARIMA is combined with the Fourier series to increase the accuracy of the prediction in [1]. However, the limitation of this model is only be done with one series of load data [1]. ARIMAX is more complex than ARIMA, it is ARIMA with explanatory variables and it is used with the holiday covariates in [3]. The classical models are difficult to apply to big data with a variety of input data types.

In recent years, in the development of artificial intelligence and deep learning, researchers are looking to apply these advanced models to electricity load forecasting. The Long short-term memory (LSTM) model works well with sequence data. In two papers [4], [5], the authors use the LSTM model for forecasting. When comparing the forecast error of the LSTM model with classical models in the papers [4], [2], the error of the LSTM model is significantly smaller. Specifically, in [4], authors used the machine learning method LSTM to forecast, they applied LSTM to the electricity dataset in Bangladesh. Bangladesh's datasets are non-stationary, so the authors removed the trend and part of seasonality before using the LSTM model. The result of that process is performance is significantly improved while removing trend gives better results than removing part of the seasonality. However, in three papers [4], [5], [2], there is no optimization of the model's hyperparameters, especially when the hyperparameters have a direct influence on the number of parameters generated then directly affect the training process and the complexity of the LSTM model.

The load forecasting problem also can be solved by many techniques which is often used in other problem and the autoencoder model is one of them. In paper [6], Kang Ke et al proposed the combination model of Stack AutoEncoder (SAE) and Gate Recurrent Units (GRU). The role of SAE in that combination model is preprocessing data and GRU is the main model to forecast electricity load. Although the result

978-1-6654-4950-2/21/$31.00 ©2021 IEEE

of that model in the experiment in that paper is better than LSTM, the SAE-GRU model is more complex.

The Convolution Neural Network model (CNN) is also the popular model in image processing problems but in the paper [7], the authors achieve good results with this model. The main idea of this paper is to use an ensemble model with the coefficients of each ingredient model is trained based on bio-inspired.

Paper [8] is the additional example about combining LSTM with the other technique in image processing problems. That technique is Global Average Pooling (GAP). Moreover, this model uses three metrics to evaluate the quality of the proposed model: sensitivity, specificity, and accuracy. The problem the authors of the paper [8] want to solve is the seizure detection problem. The paper will test two cases of condition which are perfect condition and noise conditions. In both cases, the result is very good.

There are many papers that concentrate on preprocessing datasets to improve the quality of the forecast. Paper [9] shows how to choose the meaningful features from a large set of features from the dataset. The problem that the paper wants to solve is also predicting the electrical energy consumption of the house. In the context of the development of deep learning, we have to learn how to use many features instead of only using one feature as a statistical method.

In addition, reinforcement learning (RL) has significant development because of being applied successfully in many fields. In autonomous driving, the paper [10] proposes to use the Deep Deterministic Policy Gradient algorithm to train the model on The Open Racing Car Simulator game application, where the sensors are automatically returned from the application. RL is also used in different fields when using virtual exploration in the paper [5] to optimize the policy of setting electricity pricing based on "virtual" response forecasts of customers.

A successful combination of LSTM and RL is in paper [5] solves online electricity pricing problem based on the response of the customer predicted by the LSTM model then training the RL framework to explore the response of customers whenever an electricity price is introduced. Xiangyu Konga et al. overcome the limitation of discovering the response space of customers because of time to collect all responses is very long. Moreover, with the help of RL, the problem about the number of times they can introduce the price of electricity is not restricted. The role of LSTM in that model is to predict the "response" of customers. RL is also applied in freight demand forecasting which is a problem in [11]. The role of RL in that paper is to choose the coefficient for ingredient model in ensemble model based on the current forecast results and the past forecast performance of each component model. Paper deployed forecasting in 2 intervals which are short-term and long-term.

The optimization of the model's hyperparameters was also proposed and solved when the authors of the paper [12] introduced a model which has been defined as agent, environment, action, state quantities, and reward, respectively. They used the Deep Q Network algorithm to find the best set of hyperparameters. However, the application of RL to optimize a specific model is not mentioned.

The purpose of this paper is to forecast highly accurate short-term electricity loads based on deep learning and reinforcement learning. Specifically, the load forecasting based on the combination model between LSTM and RL is proposed. LSTM model is used to transmit electricity load information as well as other information from the previous hours, previous days to the future. Moreover, the LSTM framework was created to define the environment when using the RL framework to solve the hyperparameters optimization problem. Here, the RL is used to train the hyperparameter of the LSTM model. The goal of reinforcement learning is to help the agent find out a good policy that decreases the value of forecast error.

The paper is divided into three parts: (i) Part 1 is an overview of electricity load forecasting by deep learning and reinforcement learning; (ii) Part 2 is LSTM, RL method, and proposed method combining LSTM and RL to forecast; (iii) Part 3 is the application of the proposed method to forecast national electricity load in VietNam.

II. Methodology

A. Markov decision process and Q learning algorithm

There are five qualities in the reinforcement learning problem which is modeled by the Markov decision process (MDP) through four main quantities $(\mathbb{S}, \mathbb{A}, \mathbb{P}, \mathbb{R})$. $\mathbb{S}$ is the space of state that the agent can obtain, all the actions that can be performed by the agent on the environment are in $\mathbb{A}$. A reward value will be generated immediately after performing action a to the environment at specific state s then obtain the next state s' and reward is denoted as $R(s, a, s')$ if the reward function $\mathbb{R}$ is determined. If the environment is random, performing the action a_t at state s_t can return more than one state s_{t+1} in $\mathbb{S}$. The probability in a random environment is the transition probability function $P(s, a, s') = p(s'|s, a)$.

The function evaluates the execution of an action $A \in \mathbb{A}$ at a state $S \in \mathbb{S}$ according to Bellman's equation when performing policy π_0. That function is called the action-value function:

$$\begin{aligned} Q_{\pi_0}(S, A) &= E_{\pi_0}[K_t|(s_t, a_t) = (S, A)] \\ &= E_{\pi_0}[r_t + \lambda Q_{\pi_0}(s_{t+1}, a_{t+1})|(s_t, a_t) = (S, A)] \end{aligned} \tag{1}$$

We set $K_t = Return(t)$ as the return function from time t to time L which is the length of the trajectory. The trajectory is obtained after the agent makes a decision sequentially from action at an initial time to time L. Moreover, the reward series is obtained after the trajectory $(S_0, A_0, S_1, A_1, .., S_t, A_t, ..., S_L, A_L)$ has been determined.

$$Return(t) = r_t + \beta_{t+1} r_{t+1} + \beta_{t+2} r_{t+2} + ... + \beta_L r_L = \sum_{i=t}^{L} \beta_i r_i \tag{2}$$

In Eq.(2) $\beta_i = \lambda^i$ represents the priority in finding the optimal policy, $\lambda \in (0, 1)$ is the discount factor.

Besides, the function that evaluates only one state is less commonly used. The formula of value function as Eq.(3)

$$V_{\pi_0}(S) = E_{\pi_0}[K_t|s_t = S] = E_{\pi_0}[r_t + \lambda K_{t+1}|s_t = S] \\ = E_{\pi_0}[r_t + \lambda V_{\pi_0}(s_{t+1})|s_t = S, a_t \sim \pi_0(s_t)] \quad (3)$$

A good policy is defined as making the agent perform actions to obtain the trajectory with a high return value.

However, because of the difficulty of building the model of the environment which consists of transition probability function $\mathbb{P}$ and the reward function $\mathbb{R}$, the model-based learning algorithm is less commonly used than model-free algorithm whose data are obtained by interacting with the environment. One of the most popular and effective model-free algorithms is Q learning. The formula for updating the function $Q(s, a)$ where α is learning rate and γ is the discount factor:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (4)$$

The behavior algorithm and the target algorithm are required by the off-policy algorithm to balance exploration and exploitation. Another commonly used exploration algorithm is epsilon greedy. Q learning is also an off-policy algorithm and uses the epsilon-greedy algorithm.

If the state space and action space are small, Q learning will bring a good policy. However, the Deep Q Network algorithm will be used instead of Q learning if the state space or the action space is large or continuous.

B. Long short-term memory model

The important problem when using a traditional neural network is adding layers so the number of parameters that need to be optimized will increase. However, that problem is not important in the Long Short-Term Memory (LSTM) model. The addition of LSTM cells does not affect the number of parameters. LSTM is an innovation of the Recurrent Neural Network (RNN). While RNN is simply designed with only one step from input to output by using the $tanh$ function (Figure 1), LSTM has up to three gates to process the input and LSTM also introduces a new state which stores information through LSTM cells - the cell state (Figure 2). It is possible to use LSTM to send information from previous cells to later cells. Moreover, LSTM overcomes the vanishing gradient problem.

The mission of forger gate is remove unnecessary information for that cell with range of value is (0,1):

$$f_t = sigmoid(u_f x_t + w_f h_{t-1} + b_f) \quad (5)$$

Input gate has the responsibility to add necessary information to the cell state by combining with $I_t = tanh(u_c x_t + w_c h_{t-1} + b_c)$:

$$i_t = sigmoid(u_i x_t + w_i h_{t-1} + b_i) \quad (6)$$

Output gate will combine with cell state to calculate the hidden state - output of LSTM cell:

$$o_t = sigmoid(u_o x_t + w_o h_{t-1} + b_o) \quad (7)$$

Cell state is updated through each LSTM cell:

$$c_t = f_t \otimes c_{t-1} + i_t \otimes I_t \quad (8)$$

Hidden state - output of LSTM cell :

$$h_t = o_t \otimes tanh(c_t) \quad (9)$$

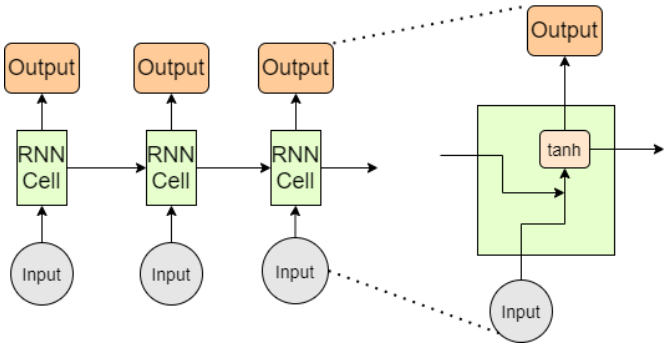

Fig. 1. Structure RNN.

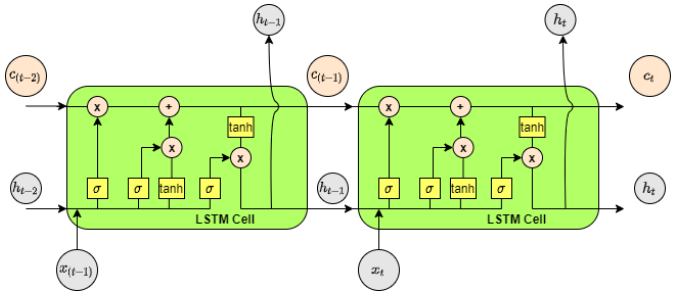

Fig. 2. Structure LSTM.

1) Some hyperparameters of LSTM.: Some hyperparameters of the LSTM model as timesteps which is the number of LSTM cells, number of units which is the number of dimensions of hidden state, optimization method, loss function.

C. Proposed method

The model combining RL and LSTM for the LSTM's hyperparameters optimization problem will have two main blocks: LSTM block and RL block. In the LSTM block, two main missions are the definition of the LSTM framework and the definition of two LSTM's hyperparameters that need to be optimized. Besides, the RL block will define the problem based on the reinforcement learning framework.

1) LSTM block:

LSTM framework: In our proposed method, the LSTM framework is important to complete our idea. Figure 3 is the definition of the LSTM framework which consists of three ingredients. Two ingredients are the dataset which is divided into the training set and the test set. The other is a set of value of $p-k$ hyperparameters where p is the number of hyperparameters of LSTM, k is the number of hyperparameters which are unknown and need to be optimized. LSTM framework not only is a combination of three ingredients but also can process that ingredient when k hyperparameters are determined then passing to the LSTM framework. LSTM framework after having all values of hyperparameters can be called a complete LSTM framework. Firstly, LSTM which contains all hyperparameters of the complete LSTM framework will be training with training data then obtain trained parameters. Secondly, we forecast electricity load based on LSTM which is just trained then compute forecast error with the test set. Therefore, the output of the LSTM framework is forecast values and forecast error.

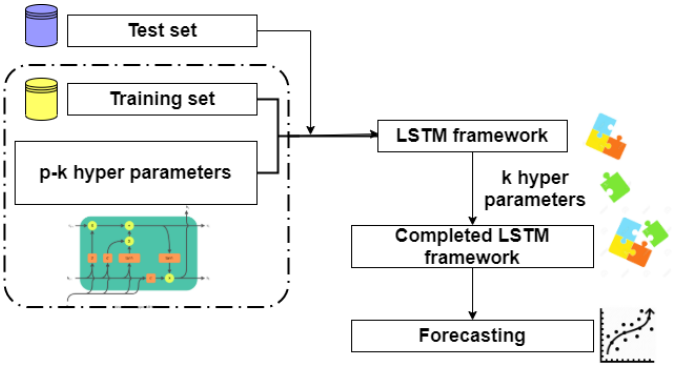

Fig. 3. Structure LSTM framework.

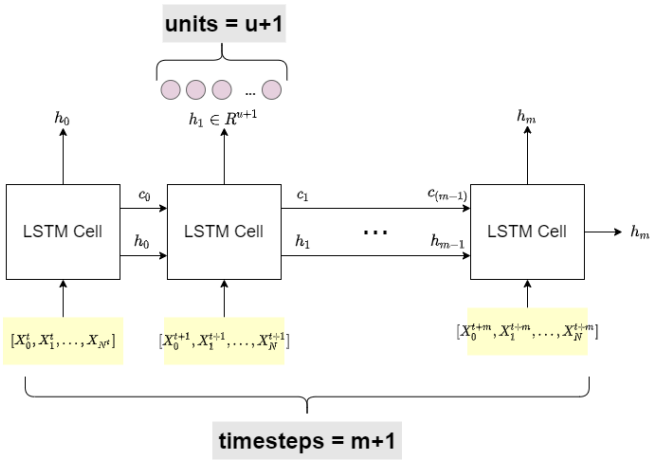

Fig. 4. Two hyperparameters LSTM.

Two hyperparameters of LSTM: Two hyperparameters optimized by reinforcement learning in our proposed method are timesteps and units, which is illustrated in Figure 4. Timesteps are the number of LSTM cells to use for the LSTM model to forecast while units are the dimension of the hidden state. In Figure 4, we use $m+1$ LSTM cells so $timesteps = m+1$ and the hidden state of each cells have $u+1$ dimensions so $units = u+1$.

2) RL block:

The proposed model for the hyperparameters optimization problem of LSTM uses a reinforcement learning framework.: There are five qualities of RL that need to be determined before optimizing LSTM's hyperparameters and they are state, agent, action, environment, and reward. These quantities are shown in Figure 5.

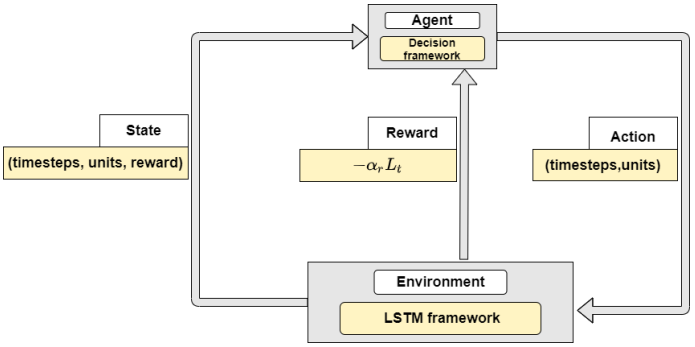

Fig. 5. RL object in optimize LSTM hyperparameters problem.

The environment that needs to be explored and be able to return the reward in the LSTM's hyperparameter optimization problem is the incomplete LSTM framework with two missing hyperparameters $timesteps$ and $units$. The action is the tuple two hyperparameters. Therefore, the action space $\mathbb{A} = T \times U$ where T is the space of $timesteps$ and U is the space of $units$. Fitting the action to the environment $a_t = (timesteps_t, units_t) \in \mathbb{A}$ is also fitting the set of hyperparameter a_t to the LSTM framework $\mathbb{F}$.

After fitting a_t to $\mathbb{F}$, the complete LSTM model has been obtained, the model will be trained with the training data, then tested on the test data and forecast $PRE = (predict)_{j=1,2,\ldots,N}$. We determined the forecast error L_t. The goal of each step of reinforcement learning is to make the high reward ($\max r_t$) while the target of the forecast is the small forecast error ($\text{argmin} L_t$), so with this proposed method and the scalar $\alpha_r > 1$, we set the reward as

$$r_t = -\alpha_r L_t = -\alpha_r MAPE(ACTUAL, PRE) = -\frac{\alpha_r}{N}\sum_{j=1}^{N}\frac{|actual_j - predict_j|}{actual_j} \quad (10)$$

The reason we add the parameter α_r is forecast error of good model with the different set of hyperparameters is commonly in small range so with $\alpha_r > 1$, which can make during training process is easier.

The best hyperparameters will be found after giving the sequence of actions. The next state s_{t+1} is generated that is the combination of current action $a_t = (timesteps_t, units_t)$ and the current reward r_t: $s_{t+1} = (timesteps_t, units_t, r_t)$.

The agent in our proposed method is the decision framework from the algorithms used to find the best set of hyperparameters. The algorithm will set the policy to choose the next action.

The value-based algorithm will be used to find the best set of hyperparameters. The function is used to evaluate is the action-value function. After training many episodes, we obtain the good Q values $Q(S, A)$ which evaluates pretty accurately when performing action A at state S. Then the good policy will be found by choosing the next action $a_{t+1} = (timesteps_{t+1}, units_{t+1}) = argmax_a Q(s_{t+1}, a)$ where $s_{t+1} = (timesteps_t, units_t, r_t) \in \mathbb{S}$.

III. Experiment and Result

A. Experiment setting

1) Dataset: The electricity load data set in Vietnam within one year is used to illustrate using reinforcement learning for optimizing hyperparameters of LSTM then apply to the electricity load forecast. Each record data have five features about the temperature at different locations, and one target feature is the total national load. Dataset is collected from January 1, 2014, to March 10, 2015. Figure 6 is hourly electricity load time series data which is divided into the training set and test set.

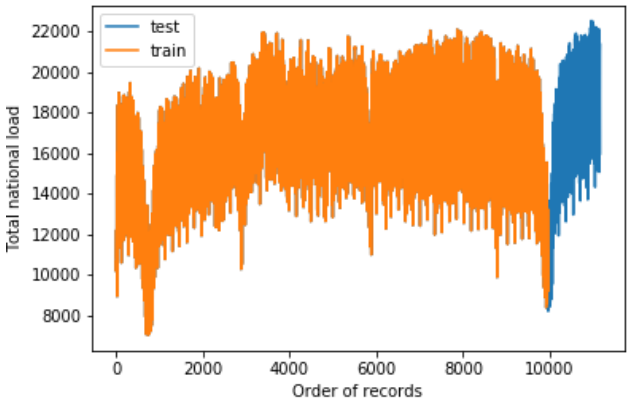

Fig. 6. Electric load data for 465 days corresponds to 11160 records.

2) LSTM framework setting:

1) Firstly, we will add new features to obtain a better forecast.
2) Fixed some hyperparameters to obtain the LSTM framework.

TABLE I
FIXED FOUR HYPERPARAMETERS OF THE LSTM FRAMEWORK.

Name of hyperparameters	Value
Optimization method	Adam
Learning rate	0.001
Loss function for neural network	Mean Absolute Error
Number of epochs	100

3) Reinforcement framework setting: We used the Deep Q Network algorithm to find the best set of hyperparameters of LSTM. We experiment our proposed method in four different cases of the number of the set of hyperparameters is $\{9, 24, 45, 90\}$.

4) Evaluate metric: Forecast error is smaller, which means the LSTM model with known all hyperparameters is a better result. As we mentioned above, the Mean Average Percentage Error (MAPE) is used to calculate the percentage of the error of the forecasted N compared to the actual value.

$$MAPE(ACTUAL, PRE) = \frac{1}{N}\sum_{j=1}^{N}\frac{|actual_j - predict_j|}{actual_j} \tag{11}$$

B. Experiment analysis and result

Because of the fixation of four hyperparameters of the LSTM model as Table 1, the LSTM framework is created. Some meaningful results are obtained when optimizing two hyperparameters based on the reinforcement learning framework.

1) After training many episodes, the policy to find out the best set of hyperparameters is simple as choosing the best action that brings the maximum Q value. We perform a sequence of action with $a_{t+1} = argmax_a Q(s_{t+1}, a)$ and after certain steps, we will obtain the best action (the best set of hyperparameters). The successful ratio is the number of the set that can lead to the best set. In Table 2, although we experiment in different cases, the successful ratio is 100% that means every set can lead to the best set.

TABLE II
EXPERIMENTAL RESULTS WITH DIFFERENT SETS OF HYPERPARAMETERS.

Number of sets	The best set	Forecast error	Successful ratio	Time (h)
9	(200,12)	8.54	9/9	2
24	(360,24)	7.61	24/24	5
45	(360,24)	7.61	45/45	10
90	(360,24)	7.61	90/90	20

2) The test data will compare with our forecast values after finding out the best set (360,24) in case 90 sets of hyperparameters and the forecast error respectively is only 7.6%. Details of forecast values are shown in Figure 7, with the blue line being the real values, the orange line being the forecast values. It is clear that the forecast line matches quite well with the real line. Because our proposed model is combined with LSTM and RL to optimize hyperparameters, the result (MAPE = 7.6%) is better than the LSTM model which is not optimized hyperparameters.

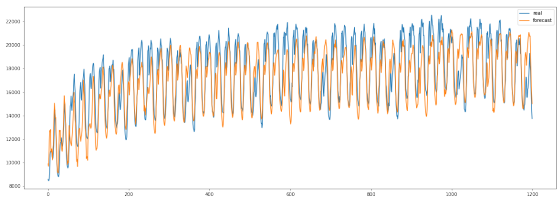

Fig. 7. Electricity load forecasting by LSTM whose hyperparameters are trained.

IV. Conclusion and discussions

This paper introduces the new proposed method for forecasting by using reinforcement learning to optimize hyperparameters of long-short term memory. The achieved results of this paper are as follows:

- Proposed the new technique for short-term forecasting.
- Applied a proposed model for electricity load forecasting.
- The performance of empirical proposed model forecasting is better than the pure LSTM model's performance.

The reason for better performance is that the proposed model uses reinforcement learning to optimize hyperparameters and when we obtain a good set of hyperparameters, the performance is apparently improved. In the future, the number of hyperparameters of the LSTM model will be researched by using reinforcement learning.
In addition, more applications using combining reinforcement and LSTM are considered.

References

[1] M. Eljazzar and E. Hemayed, "Enhancing electric load forecasting of arima and ann using adaptive fourier series," 01 2017, pp. 1–6, doi: 10.1109/CCWC.2017.7868457.

[2] S. Siami Namini, N. Tavakoli, and A. Siami Namin, "A comparison of arima and lstm in forecasting time series," 12 2018, pp. 1394–1401, doi: 10.1109/ICMLA.2018.00227.

[3] L.-Y. Chiu, D. J. Arcega Rustia, C.-Y. Lu, and T.-T. Lin, "Modelling and forecasting of greenhouse whitefly incidence using time-series and arimax analysis," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 196–201, 2019, 6th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896319324401

[4] H. Al-Shaikh and A. Zubair, "Electric load forecasting with hourly precision using long short-term memory networks," 02 2019, doi: 10.1109/ECACE.2019.8679244.

[5] X. Kong, D. Kong, J. Yao, L. Bai, and J. Xiao, "Online pricing of demand response based on long short-term memory and reinforcement learning," *Applied Energy*, vol. 271, p. 114945, 2020, https://doi.org/10.1016/j.apenergy.2020.114945.

[6] K. Ke, S. Hongbin, Z. Chengkang, and C. Brown, "Short-term electrical load forecasting method based on stacked auto-encoding and gru neural network," *Evolutionary Intelligence*, vol. 12, 09 2018, doi: 10.1007/s12065-018-00196-0.

[7] N. Parida, D. Mishra, K. Das, N. K. Rout, and G. Panda, "On deep ensemble cnn–sae based novel agro-market price forecasting," *Evolutionary Intelligence*, vol. 14, 06 2021, doi: 10.1007/s12065-020-00466-w.

[8] D. K. Thara, B. G. Premasudha, R. S. Nayak, T. V. Murthy, G. A. Prabhu, and N. Hanoon, "Electroencephalogram for epileptic seizure detection using stacked bidirectional lstm-gap neural network," *Evolutionary Intelligence*, vol. 14, 06 2021, doi: 10.1007/s12065-020-00459-9.

[9] L. M. Candanedo, V. Feldheim, and D. Deramaix, "Data driven prediction models of energy use of appliances in a low-energy house," *Energy and Buildings*, vol. 140, pp. 81–97, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378778816308970

[10] S. Wang, D. Jia, and X. Weng, "Deep reinforcement learning for autonomous driving," 11 2018, preprint: arXiv:1811.11329.

[11] L. Al Hajj Hassan, H. S. Mahmassani, and Y. Chen, "Reinforcement learning framework for freight demand forecasting to support operational planning decisions," *Transportation Research Part E: Logistics and Transportation Review*, vol. 137, p. 101926, 2020, https://doi.org/10.1016/j.tre.2020.101926. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1366554519315169

[12] H. Jomaa, J. Grabocka, and L. Schmidt-Thieme, "Hyp-rl : Hyperparameter optimization by reinforcement learning," 06 2019, preprint: arXiv:1906.11527.