Prediction of Sea Surface Temperature Using Long Short-Term Memory

Qin Zhang, Hui Wang, Junyu Dong, *Member, IEEE*, Guoqiang Zhong, *Member, IEEE*, and Xin Sun, *Member, IEEE*

***Abstract*— This letter adopts long short-term memory (LSTM) to predict sea surface temperature (SST), and makes short-term prediction, including one day and three days, and long-term prediction, including weekly mean and monthly mean. The SST prediction problem is formulated as a time series regression problem. The proposed network architecture is composed of two kinds of layers: an LSTM layer and a full-connected dense layer. The LSTM layer is utilized to model the time series relationship. The full-connected layer is utilized to map the output of the LSTM layer to a final prediction. The optimal setting of this architecture is explored by experiments and the accuracy of coastal seas of China is reported to confirm the effectiveness of the proposed method. The prediction accuracy is also tested on the SST anomaly data. In addition, the model's online updated characteristics are presented.**

***Index Terms*— Long short-term memory (LSTM), prediction, recurrent neural network (RNN), sea surface temperature (SST), SST anomaly.**

I. Introduction

SEA surface temperature (SST) is an important parameter in the energy balance system of the earth's surface, and it is also a critical indicator to measure the heat of sea water. It plays an important role in the process of the earth's surface and atmosphere interaction. Sea occupies three quarters of the global area; therefore, SST has an inestimable influence on the global climate and the biological systems. The prediction of SST is also important and fundamental in many application domains, such as ocean weather and climate forecast, offshore activities like fishing and mining, ocean environment protection, ocean military affairs, and so on. It is significant in science research and application to predict the accurate temporal and spatial distribution of SST. However, the accuracy of its prediction is always low due to many uncertain factors especially in coastal seas. This problem is especially obvious in coastal seas.

Many methods have been published to predict SST. These methods can be generally classified into two categories [1]. One is based on physics, which is also known as the numerical model. The other is based on data, which is also called the data-driven model. The former tries to utilize a series of differential equations to describe the variation of SST, which is usually sophisticated and demands increasing computational effort and time. In addition, numerical model differs in different sea areas, whereas the latter tries to learn the model from data. Some learning methods have been used, such as linear regression [2], support vector machines [3], neural network [1], and so on.

This letter employs the latter to predict SST, which uses long short-term memory (LSTM) to model the time series of SST data. Long short-term memory is a special kind of recurrent neural network (RNN), which is a class of artificial neural network where connections between units form a directed cycle. This creates an internal state of the network, which allows it to exhibit dynamic temporal behavior. Unlike feed-forward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs [4]. However, vanilla RNN suffers a lot about vanishing or exploding gradient problem, which cannot solve the long-term dependence problem. And it is very difficult to train. While LSTM introduces the gate mechanism to prevent back-propagated errors from vanishing or exploding, which has been subsequently proved to be more effective than conventional RNNs [5].

In this letter, an LSTM-based method is proposed to predict SST. There are two main contributions. First, an LSTM-based network is properly designed with a full-connected layer to form a regression model for SST prediction. The LSTM layer is utilized to model the temporal relationship among SST time series data. A full-connected layer is applied to map the output of the LSTM layer to the final prediction result. Second, SST change is relatively stable in ocean, while it is more fluctuated in coastal seas. So the SST values of Baohai coastal seas are adopted in the experiments, and the prediction results beyond the existing methods are reported to confirm the effectiveness of the proposed method.

The remainder of this letter is organized as follows. Section II gives the problem formulation and describes the proposed method in detail. Experimental results on Bohai SST Data Set, which is chosen from NOAA OI SST V2 High-Resolution Data Set, are reported in Section III. Finally, Section IV concludes this letter.

II. Methodology

A. Problem Formulation

Usually, the sea surface can be divided into grids according to the latitude and longitude. Each grid will have a value at

Manuscript received March 14, 2017; revised June 11, 2017 and July 21, 2017; accepted July 25, 2017. Date of publication August 11, 2017; date of current version September 25, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 41576011, Grant 61403353, and Grant 61401413 and in part by the International Science and Technology Cooperation Program of China under Grant 2014DFA10410. *(Corresponding author: Junyu Dong.)*

Q. Zhang is with the Department of Computer Science and Technology, Ocean University of China, Qingdao 266100, China, and also with the Department of Science and Information, Agriculture University of Qingdao, Qingdao 266109, China.

H. Wang is with the College of Oceanic and Atmospheric Sciences, Ocean University of China, Qingdao 266100, China.

J. Dong, G. Zhong, and X. Sun are with the Department of Computer Science and Technology, Ocean University of China, Qingdao 266100, China (e-mail: dongjunyu@ouc.edu.cn).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2017.2733548

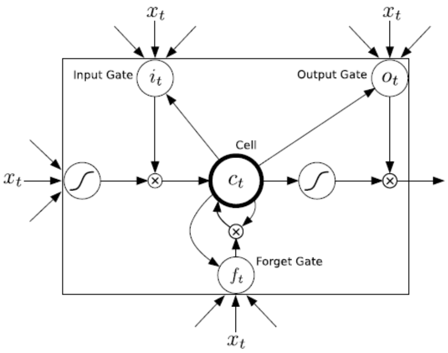

Fig. 1. Stucture of LSTM cell [6].

an interval of time. Then the SST values can be organized as 3-D grids. The problem is how to predict the future value of SST according this 3-D SST grid.

To make the problem simpler, suppose the SST values from one single grid is taken during the time, it is a sequence of real values. If a model can be built to capture the temporal relationship among data, then the future values can be predicted according to the historical values. Therefore, the prediction problem at this single grid can be formulated as a regression problem: if k days' SST values are given, what are the SST values for the $k+1$ to $k+l$ days? Here, l represents the length of prediction.

B. Long Short-Term Memory

To capture the temporal relationship among time series data, LSTM is adopted. LSTM was first proposed by Hochreiter and Schmidhuber [6] in 1997. It is a specific RNN architecture that was designed to model sequences and can solve the long-range dependences more accurately than conventional RNNs. LSTM can process a sequence of input and output pairs $\{(x_t, y_t)\}_{t=1}^{n}$. For current time step with the pair (x_t, y_t), the LSTM cell takes a new input x_t and the hidden vector h_{t-1} from the last time step, and then produces an estimate output $\hat{y}_t$ also with a new hidden vector h_t and a new memory vector m_t. Fig. 1 shows the structure of an LSTM cell. The whole computation can be defined by a series of equations as follows [7]:

$$\begin{aligned} i_t &= \sigma(W^i H + b^i) \\ f_t &= \sigma(W^f H + b^f) \\ o_t &= \sigma(W^o H + b^o) \\ c_t &= \tanh(W^c H + b^c) \\ m_t &= f_t \odot m_{t-1} + i_t \odot c_t \\ h_t &= \tanh(o_t \odot m_t) \end{aligned} \tag{1}$$

where σ is the sigmoid function, W^i, W^f, W^o, and W^c in $\mathbb{R}^{d\times 2d}$ are the recurrent weight matrices, and b^i, b^f, b^o, and b^c are the corresponding bias terms. H in $\mathbb{R}^{2d}$ is the concatenation of the new input x_t and the previous hidden vector h_{t-1}

$$H = \begin{bmatrix} Ix_t \\ h_{t-1} \end{bmatrix}. \tag{2}$$

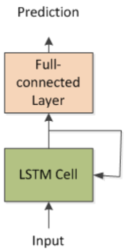

Fig. 2. Basic LSTM block.

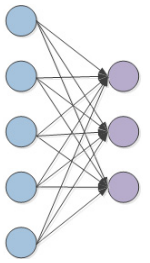

Fig. 3. Full-connected layer.

The key to LSTM is the cell state, i.e., memory vector m and m' in (1), which can remember long-term information. The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates. The gates in (1) are i, f, o, and c, representing an *input gate*, a *forget gate*, an *output gate*, and a *control gate*. The input gate can decide how much input information enters the current cell. The forget gate can decide how much information be forgotten for the previous memory vector m_{i-1}, while the control gate can decide to write new information into the new memory vector m_i modulated by the input gate. The output gate can decide what information will be output from the current cell.

Following the work of [8], we also use a whole function LSTM() as shorthand for (1):

$$(h', m') = \text{LSTM}\left(\begin{bmatrix} Ix_i \\ h_{i-1} \end{bmatrix}, m, W\right) \tag{3}$$

where W concatenates the four weight matrices W^i, W^f, W^o, and W^c.

C. Basic LSTM Blocks

LSTM is combined with a full-connected layer to build a basic LSTM block. Fig. 2 shows the structure of a basic LSTM block. There are two basic neural layers in a block. The LSTM layer can capture the temporal relationship, i.e., the regular variation among the time series SST values. While the output of the LSTM layer is a vector i.e., the hidden vector of the last time step, a full-connected layer is used to make a better abstraction and combination for the output vector, and reduces its dimensionality, meanwhile maps the reduced vector to the final prediction. Fig. 3 shows a full-connected layer. The computation can be defined as follows:

$$\begin{aligned} (h_i, m_i) &= \text{LSTM}\left(\begin{bmatrix} \text{Iinput} \\ h_{i-1} \end{bmatrix}, m, W\right) \\ \text{prediction} &= \sigma(W^{\text{fc}} h_l + b^{\text{fc}}) \end{aligned} \tag{4}$$

where the definition of function LSTM() is as (3), h_l is the hidden vector in the last time step of LSTM, W^{fc} is

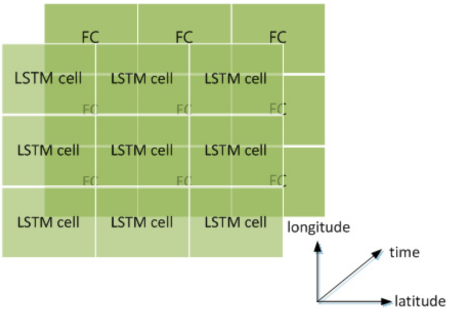

Fig. 4. Network architecture.

the weight matrices in a full-connection layer, and b^{fc} is the corresponding bias terms.

This kind of block can predict future SST of a single grid, according to all the previous SST values of this grid. But it is still not enough. Prediction of SST of an area is needed. So the basic LSTM blocks can be assembled to construct the whole network.

D. Network Architecture

Fig. 4 shows the architecture of the network. It is like a cuboid: the x-axis stands for latitude, the y-axis stands for longitude, and the z-axis is time direction. Each grid corresponds to a grid in real data. Actually, the grids in the same place along the time axis form a basic block. We omit the connections between layers for clarity.

III. Results and Discussion

A. Study Area and Data

We use NOAA high-resolution SST data provided by the NOAA/OAR/ESRL PSD, Boulder, Colorado, USA, from their Web site at http://www.esrl.noaa.gov/psd/ [9]. This data set contains SST daily mean values, SST daily anomalies, SST weekly mean, and monthly mean values from September 1981 to November 2016 (12 868 days in total), and covers the global ocean from 89.875S to 89.875N, 0.125E to 359.875E, which is 0.25° latitude multiplied by 0.25° longitude global grid (1440 × 720).

It is known that the temperature varies relatively stable in far ocean, and fluctuates more greatly in coastal seas. So the coastal seas near China are focused to evaluate the proposed method. The Bohai Sea is the innermost gulf of the Yellow Sea and Korea Bay on the coast of northeastern and northern China. It is approximately 78 000 km^2 in area and its proximity to Beijing, the capital of China, makes it one of the busiest seaways in the world [10]. Bohai sea covers from 37.07N to 41N, 117.35E to 121.10E. We take the corresponding subset to the Bohai Sea from the data set mentioned above to form a 16 × 15 grid, which contains a total of 12 868 daily values, named the Bohai SST data set. It contains four data subsets. The Bohai SST daily mean data set and the Bohai SST daily anomaly data set are used for daily prediction. The Bohai SST weekly mean data set and the Bohai SST monthly mean data set are used for weekly and monthly prediction.

B. Experimental Setup

Since the SST prediction is formulated as a sequence prediction problem, i.e., using previous observations to predict the future, the duration the previous observations are to be used to predict the future should be determined. Of course, the longer the length is, the better the prediction will be. Meanwhile, more computation will be needed. Here, the length of the previous sequence is set to four times of the length of prediction according to the characteristics of the periodical change of temperature data. In addition, there are still other important values to be determined: the number of layers for the LSTM layer l_r and the full-connected layer l_{fc}, which will determine the whole structure of the network. Also the corresponding number of hidden units denoted by $units\_r$ should be determined together.

According to these aspects mentioned above, we first design a simple but important experiment to determine the critical values for l_r, l_{fc}, and $units\_r$, using the basic LSTM block to predict the SST for a single location. Then, we evaluate the proposed method on area SST prediction for Bohai Sea.

Once the structure of the network is determined, there are still other critical things to be determined in order to train the network, i.e., the activation function, the optimization method, the learning rate, the batch size, and so on. The basic LSTM block uses logistic sigmoid and hyperbolic tangent as an activation function. Here, we use an ReLU activation function for it is easy to optimize and is not saturated. The traditional optimization method for a deep network is stochastic gradient descent (SGD), which is the batch version of gradient descent. The batch method can speed up the convergence of network training. Here, we adopt the Adagrad optimization method [11], which can adapt the learning rate to the parameters, performing larger updates for infrequent and smaller updates for frequent parameters. Dean *et al.* [12] have found that Adagrad improved the robustness of SGD greatly and used it for training large-scale neural networks. We set the initial learning rate as 0.1, and the batch size as 100 in the following experiments.

The division of training set, validation set, and test set is as follows. The data from September 1981 to August 2012 (11 323 days in total) are used as training set, the data from September 2012 to October 2012 (122 days in total) are the validation set, and the data from January 2013 to December 2015 (1095 days in total) are the test set. We will test for one week (7 days) and one month (30 days) to evaluate the prediction performance. The data of 2016 (328 days in total) is reserved for another comparison.

Results of another two regression models, i.e., support vector regression (SVR) [13] and multilayer perceptron regression (MLPR) [14], for SST prediction are given for the purpose of comparison. SVR is one of the most popular regression models in recent years, which has achieved good results in many application domains, while MLPR is a typical artificial neural network for regression task. We run the experiments under the environment of Intel Core2 Quad CPU Q9550 @2.83-GHz, 6G RAM, Ubuntu 16.10 64-b operating system, and Python 2.7. The proposed network is implemented by Keras [15]. SVR and MLPR are implemented by Scikit-learn [16].

The performance evaluation of SST prediction is a fundamental issue. In this letter, the root mean squared error (RMSE) is adopted, which is one of the most common measurement used as the evaluation metric to measure the effectiveness of different methods. Apparently, the smaller the

TABLE I

PREDICTION RESULTS (RMSE) ON FIVE LOCATIONS WITH DIFFERENT $Units_r$S

$units\_r$	p_1	p_2	p_3	p_4	p_5
1	0.1595	0.1171	0.2690	0.2988	0.2626
2	0.1589	0.1137	0.2569	0.2909	0.2695
3	0.2075	0.0923	0.2580	0.2819	0.2606
4	0.2152	0.0918	0.2349	0.2752	0.2672
5	**0.1280**	**0.0914**	**0.2310**	0.2723	**0.2362**
6	0.1353	0.0922	0.2454	**0.2646**	0.2468

TABLE II

PREDICTION RESULTS (RMSE) ON FIVE LOCATIONS WITH DIFFERENT l_rS

l_r	p_1	p_2	p_3	p_4	p_5
1	**0.1280**	**0.0914**	**0.2310**	**0.2723**	**0.2362**
2	0.1288	0.1153	0.2500	0.2730	0.2496
3	0.3659	0.0950	0.2656	0.2732	0.3334

TABLE III

PREDICTION RESULTS (RMSE) ON FIVE LOCATIONS WITH DIFFERENT kS

l_{fc}	p_1	p_2	p_3	p_4	p_5
1[3]	**0.1280**	**0.0914**	**0.2310**	**0.2723**	**0.2362**
2[3,3]	0.2838	0.0945	0.2880	0.2724	0.2461
2[6,7]	0.3660	0.2605	0.4655	0.2730	0.3355

TABLE IV

PREDICTION RESULTS (AREA AVERAGE RMSE) ON THE BOHAI SST DATA SET

Methods	Daily		Weekly	Monthly
	one day	three days	one week	one month
SVR	0.3998	0.6158	0.4716	0.6538
MLPR	0.6633	0.8215	0.6919	0.8360
LSTM network	**0.0767**	**0.1775**	**0.3844**	**0.3928**

RMSE is, the better the performance is. Here, RMSE can be regarded as an absolute error. And for area prediction, the area average RMSE is used.

C. Determination of Parameters

We randomly choose five locations in the Bohai daily mean SST data set denoted as $p_1, p_2, \ldots, p_5$ to predict three days' SST values with a half-month (15 days) length of the previous sequence. First, we fix l_r and l_{fc} as 1 and $units_$fc as 3, and choose a proper value for $units\_r$ from $\{1, 2, 3, 4, 5, 6\}$. Table I shows the results on five locations with different values of $nunits\_r$. The boldface items in the table represent the best performance, i.e., the smallest RMSE. It can be seen from the results that the best performance occurs when $units\_r = 6$.

In this experiment, the best performance occurs when $units\_r = 5$ in four locations p_1, p_2, p_3, and p_5, while at p_4, the best performance occurs when $units\_r = 6$. We can see that the difference in RMSE is not too significant. So, in the following experiments, we set $units\_r$ as 5.

Then, we also use the SST sequences from the same five locations to choose a proper value for l_r from $\{1, 2, 3\}$. The other two parameters are set by $unit\_r = 5$ and $l_{\text{fc}} = 1$. Table II shows the results on five locations with different values of l_r. The boldface items in the table represent the best performance. It can be seen from the results that the best performance occurs when $l_r = 1$. The reason may be the increasing weights with increasing recurrent LSTM layers. In this case, the training data are not sufficient enough to learn so many weights. Actually, experiences in previous study show that the recurrent LSTM layer is not the more the better. And during the experiments, we find that the more the LSTM layers are, the more likely to get unstable results and the more training time to be needed. Therefore, in the following experiments, we set l_r as 1.

Finally, we still use the SST sequences from the same five locations to choose a proper value for l_{fc} from $\{1, 2\}$. Table III shows the results with different l_{fc}s. The numbers in the square brackets stand for the number of the hidden units. The boldface items in the table represent the best performance. It can be seen from the results that it achieves the best performance when $l_{\text{fc}} = 1$. The reason may be the same: more layers mean more weights to be trained and more computation it needs. Therefore, in the following experiments, we set l_{fc} as 1, and the number of its hidden units is set to the same value as the prediction length.

To summarize, the number of LSTM layers and full-connected layers are set to be 1. The number of the neurons in the full-connected layer is set the same as the prediction length l. The number of the hidden units in the LSTM layer is chosen in an empirical value range $\lfloor (l/2), 2l \rfloor$. More hidden units require more computational time; thus, the number needs to be balanced in the application.

D. Results and Discussion

We use the Bohai SST data set to do this experiment and compare the proposed method with two classical regression methods SVR and MLPR. Specifically, the Bohai SST daily mean data set and the daily anomaly data set are used for one-day and three-days short-term prediction. The Bohai SST weekly mean and monthly mean data sets are used for one-week and one-month long-term prediction. The setting is as follows. For the short-term prediction of the LSTM network, we set $k = 10, 15, 30, 120$ for $l = 1, 3, 7, 30$, respectively, and $l_r = 1$, $units\_r = 6$, $l_{\text{fc}} = 1$. For the long-term prediction of the LSTM network, we set $k = 10$, $units\_r = 3$, $l_r = 1$ and $l_{\text{fc}} = 1$. For SVR, we use the RBF kernel and set the kernel width $\sigma = 1.6$, which is chosen by fivefold cross validation on the validation set. For MLPR, we use a three-layer perceptron network, which includes one hidden layer. The number of hidden units is the same as the setting of the LSTM network for fair comparison.

Table IV shows the results of daily short-term prediction and weekly, monthly long-term prediction. The boldface items in the table represent the best performance, i.e., the smallest area average RMSE. We also test the prediction performance with respect to the SST daily anomalies shown in Table V. It can be seen from the results that the LSTM network achieve the best prediction performance. In addition, Fig. 5 shows the prediction result at one location using different methods. In order to see the results clearly, we only show the prediction results for one year from January 1, 2013 to December 31, 2013, which is the first year of the test set.

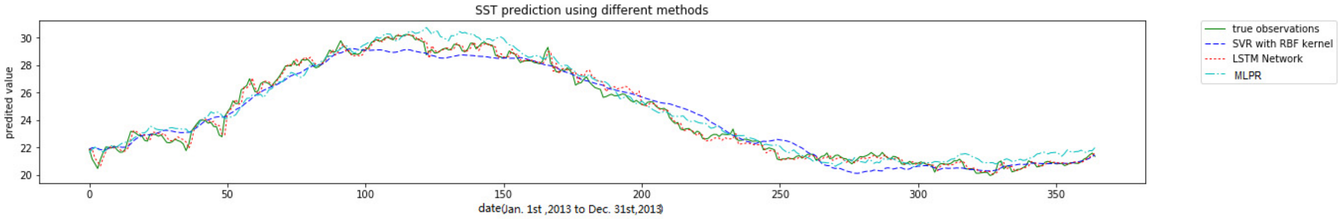

Fig. 5. SST Three-Days' Prediction at One Location Using Different Methods.

TABLE V

PREDICTION RESULTS (AREA AVERAGE RMSE) ON THE BOHAI SST DAILY ANOMALY DATA SET

Methods	Daily	
	one day	three days
SVR	0.3277	0.5266
MLPR	0.9386	0.7518
LSTM network	**0.3110**	**0.4857**

TABLE VI

PREDICTION RESULTS(AREA AVERAGE RMSE) ON BOHAI SST DAILY DATA SET IN 2016

Model	Prediction of 2016	
	one day	three days
original	0.1346	0.2145
updated	**0.0899**	**0.1843**

Green solid line represents the true value. Red dotted line represents the prediction results of the LSTM network. Blue dashed line represents the prediction results of SVR with the RBF kernel. And cyan dashed-dotted line represents the prediction results of the MLPR.

E. Online Model Update

In this experiment, we want to show the online characteristics of the proposed method. We have SST values of 328 days in 2016. We refer to the above-trained model as the *original model*, and use this model to predict the SST values of 2016. Based on the original model, we continue to train the model adding three-years' SST observations data of 2013, 2014, and 2015, and get a new model called *updated model*. Table VI shows the results of SST prediction for 2016 using these two different models. As is expected, the updated model performs the best.

This shows a kind of online characteristics of the proposed method: performing prediction, collecting true observations, feeding the true observations back into the model to update it, and so on. However, other regression models, like SVR, do not have such characteristics: when collecting new observations, the model could only be retrained from scratch, which will waste additional computing resources.

IV. CONCLUSION

In this letter, the prediction of SST is formulated as a time series regression problem, and an LSTM-based network is proposed to model the temporal relationship of SST to predict the future value. The proposed network utilizes the LSTM layer to model the time series data, and full-connected layer to map the output of the LSTM layer to the final prediction. The optimal setting of this architecture is explored through experiments and the prediction performance of coastal seas of China is reported to confirm the effectiveness of the proposed method. Also the prediction performance is tested on the SST anomaly data. In addition, the online update characteristics of the proposed method are shown. Once the predicted SST values in the future are obtained, they can be used in many application aspects, such as the prediction of ocean front and abnormal event, and so on.

Furthermore, the proposed network is independent of the spatial and temporal resolution of data. If another resolution prediction is required, all we need is to provide enough training data. Weekly and monthly mean SST data are also used to test the proposed method in our experiments. It should be noted that the most critical factor is the size of training data. As for other kind of data like seasonal or yearly data, there may not be enough training samples in our method.

REFERENCES

[1] K. Patil, M. C. Deo, and M. Ravichandran, "Prediction of sea surface temperature by combining numerical and neural techniques," *J. Atmos., Ocean. Technol.*, vol. 33, no. 8, pp. 1715–1726, 2016.

[2] J.-S. Kug, I.-S. Kang, J.-Y. Lee, and J.-G. Jhun, "A statistical approach to Indian Ocean sea surface temperature prediction using a dynamical ENSO prediction," *Geophys. Res. Lett.*, vol. 31, no. 9, pp. 399–420, 2004.

[3] I. D. Lins *et al.*, "Sea surface temperature prediction via support vector machines combined with particle swarm optimization," in *Proc. 10th Int. Probab. Safety Assessment Manage. Conf.*, vol. 10. Seattle, Washington, Jun. 2010.

[4] Wikipedia. *Recurrent Neural Network*. accessed on Feb. 15, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Recurrent_neural_network

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

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

[7] A. Graves. (2013). "Generating sequences with recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1308.0850

[8] N. Kalchbrenner, I. Danihelka, A. Graves. (2015). "Grid long short-term memory." [Online]. Available: https://arxiv.org/abs/1507.01526

[9] N. ESRL. *NOAA OI SST V2 High Resolution Dataset*. accessed on Feb. 15, 2017. [Online]. Available: http://www.esrl.noaa.gov/psd/data/gridded/data.noaa.oisst.v2.highres.html

[10] Wikipedia. Bohai Sea. Accessed on Feb. 15, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Bohai_Sea

[11] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 7, pp. 257–269, 2010.

[12] J. Dean *et al.*, "Large scale distributed deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1232–1240.

[13] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Inf. Process. Lett. Rev.*, vol. 11, no. 10, pp. 203–224, 2007.

[14] D. Rumelhart and J. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Foundations*. Cambridge, MA, USA: MIT Press, 1986.

[15] F. Chollet *et al.*, (2015). *Keras*. [Online]. Available: https://github.com/fchollet/keras

[16] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.