Prediction of Sea Surface Temperature using Long Short-Term Memory 基于LSTM的海洋温度预测

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

arXiv:1705.06861v1 [cs.CV] 19 May 2017

Abstract—**This letter adopts long short-term memory(LSTM) to predict sea surface temperature(SST), which is the first attempt, to our knowledge, to use recurrent neural network to solve the problem of SST prediction, and to make one week and one month daily prediction. We formulate the SST prediction problem as a time series regression problem. LSTM is a special kind of recurrent neural network, which introduces gate mechanism into vanilla RNN to prevent the vanished or exploding gradient problem. It has strong ability to model the temporal relationship of time series data and can handle the long-term dependency problem well. The proposed network architecture is composed of two kinds of layers: LSTM layer and full-connected dense layer. LSTM layer is utilized to model the time series relationship. Full-connected layer is utilized to map the output of LSTM layer to a final prediction. We explore the optimal setting of this architecture by experiments and report the accuracy of coastal seas of China to confirm the effectiveness of the proposed method. In addition, we also show its online updated characteristics.**

Index Terms—**Sea surface temperature, prediction, recurrent neural network, long short-term memory.**

I. INTRODUCTION

SEA surface temperature, SST for short, is an important parameter in the energy balance system of the earth's surface, and it is also a critical indicator to measure the sea water heat. 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 inestimable influence on the global climate and the biological systems. In recent years, people focus more and more on sea surface temperature. The prediction of SST becomes a hot research increasingly. It is also an important and fundamental problem in many application domains such as forecasting ocean weather and climate, offshore activities like fishing and mining, ocean environment protection, ocean military affairs, etc. It is significant in science research and application to predict accurate temporal and spatial distribution for SST. However, its prediction accuracy is always low because of many uncertain factors especially in coastal seas.

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

Q. Zhang is 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.

Many methods have been published to predict the sea surface temperature. These methods can be generally classified into two categories according to the different point of view to create models [1]. One is based on physics, also known as numerical model. The other is based on data, also called 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. While the latter tries to learn the model from data. Some learning methods were used such as linear regression [2], Support Vector Machines [3], Neural Network [1], etc. 这些背景知识可以借鉴，目前存在的方式两类，第一类是基于物理值，计算复杂，第二类是基于数据驱动。

This letter focuses on the second way to predict SST, which uses long short-term memory (LSTM) to model 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 feedforward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs [4]. However, vanilla RNN is difficult to train and suffers a lot about vanishing or exploding gradient problem, which can not solve the long-term dependency problem. While LSTM introduces gate mechanism to prevent backpropagated errors from vanishing or exploding, which has been subsequently proved to be more effective than conventional RNNs [5].

In this letter, a LSTM based prediction method for SST is proposed. Our main contributions are twofold: 1) a LSTM based network is properly designed with full connected layer to form a regression model for SST prediction. LSTM layer is utilized to catch the temporal relationship among SST time series data. Full connected layer is utilized to map the output of LSTM layer to a final prediction. 2) SST changes relatively stable in ocean, while more fluctuated in coastal seas. We focus on the latter, and report the prediction accuracy beyond the existing methods, which confirms 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 Dataset, which is chosen from NOAA OI SST V2 High Resolution Dataset are reported in Section III. Finally, Section IV concludes this letter.

II. METHODOLOGY

A. Problem formulation

Usually according to the latitude and longitude the sea surface can be divided into grids. Each grid will have a value at every time interval. Then the SST values can be organized as three dimensional grids. The problem is how to predict the future value of SST given this 3D SST grid.

Suppose we take the SST values from one single grid along all the time. It is a time series real values. If we can build a model to capture the temporal relationship among data, then we can predict the future values given the historical values. So the prediction problem can be formulated as a regression problem: give k days' SST values, what are the SST values for the $k+1$ to $k+l$ days? Here l represents the prediction length.

B. Long short-term memory

To capture the temporal relationship among time series data, we adopt LSTM to do this job. This subsection introduces LSTM briefly.

LSTM was first proposed by Hochreiter in 1997 [6]. It is a specific recurrent neural network architecture that was designed to model sequences and their long-range dependencies more accurately than conventional RNNs. LSTM can process a sequence of input and output pairs $(x_i, y_i)_{i=1}^n$. For each pair (x_i, y_i), the LSTM cell takes a new input x_i and the hidden vector h_{i-1} from the last time step, then produces an estimate output $\hat{y}_i$ for the target output y_i given all the previous input sequence $x_1, x_2, \cdots, x_i$ also with a new hidden vector h_i and a new memory vector m_i. Fig. 1 shows the structure of a LSTM cell. The whole computation can be defined by a series of equations as follows [7]:
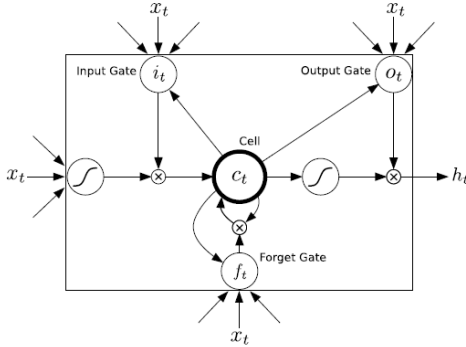

Fig. 1: Stucture of LSTM cell [6]

$$
\begin{aligned}
i &= \sigma(W^i H + b^i) \\
f &= \sigma(W^f H + b^f) \\
o &= \sigma(W^o H + b^o) \\
c &= tanh(W^c H + b^c) \\
m' &= f \odot m + i \odot c \\
h' &= tanh(o \odot m')
\end{aligned}
\tag{1}
$$

where σ is the sigmoid function, W^i, W^f, W^o, W^c in $\mathbb{R}^{d \times 2d}$ are the recurrent weight matrices, and b^i, b^f, b^o, b^c are the

corresponding bias terms. H in $\mathbb{R}^{2d}$ is the concatenation of the new input x_i and the previous hidden vector h_{i-1}:

$$
H = \left[\begin{array}{c} Ix_i \\ h_{i-1} \end{array} \right]
\tag{2}
$$

The key to LSTM is the cell state, i.e. memory vector m and m' in Equation (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 Equation (1) are i, f, o, c, representing *input gate*, *forget gate*, *output gate* and a *control gate*. Input gate can decide how much input information enter the current cell. 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. Output gate can decide what information will be output from current cell.

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

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

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

C. Basic LSTM blocks

We combine LSTM with 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. LSTM layer can capture the temporal relationship, i.e. the regulate variation among the time series SST values. While the output of LSTM layer is a vector i.e. the hidden vector of the last time step, we use a full-connected layer to make a better abstraction and combination for the output vector, and reduce its dimensionality meanwhile map the reduced vector to a final prediction. Fig. 3 shows a full-connected layer. The computation can be defined as follows:

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

where the definition of function $LSTM()$ is as Eqation(3), h_l is the hidden vector in the last time step of LSTM, W^{fc} is the weight matrices in full-connection layer, and b^{fc} is the corresponding bias terms.

This kind of block can predict future SST for a single grid, given all the historical SST values of this grid. But it's still not enough. We need to predict SST for an area. So we can use the basic LSTM blocks to construct the whole network.

D. Network architecture

Fig. 4 shows the architecture of the network. It's like a cuboid: the x axis stands for latitude, the y axis stands for longitude, and the z axis is time direction. Each grid is corresponding 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. EXPERIMENTAL RESULTS AND DISCUSSIONS

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 daily values from 1981/09 to 2016/11 (12868 days total), and covers the global ocean from 89.875S to 89.875N, 0.125E to 359.875E, which is 0.25 degree latitude multiplied by 0.25 degree longitude global grid (1440x720).

As we all know, the temperature varies relatively stable in far ocean, while fluctuates more greatly in coastal seas. So we focus on the coastal seas near China to evaluate the proposed method. Bohai sea is the innermost gulf of the Yellow Sea and Korea Bay on the coast of Northeastern and North 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

Fig. 2: Structure of a basic LSTM block

Fig. 3: A full-connected layer

Fig. 4: Network architecture

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 dataset mentioned above to form a 16 by 15 grid and contains a total of 12868 daily values, named Bohai SST Dataset.

B. Experimental Setup

Since we formulate the SST prediction as a sequence prediction problem, i.e. using historical observations to predict the future, we should determine how long the historical observations are to be used to predict the future. Of course the longer the length is, the better the prediction will be, while the more computation it will need. Here we use about 4 times of the prediction length to be the length of the historical observations 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 LSTM layer l_r and 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 position. Then we evaluate the proposed method on area SST prediction for Bohai sea.

Once we determine the structure of the network, there are still other critical things to be determined in order to train the network, i.e. the optimization method, the learning rate, the batch size, etc. The traditional optimization method for 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 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 greatly improved the robustness of SGD 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 are as follows. The data from 1981 to 2012.8 (11323 days) is used as training set, the data from 2012.9 to 2012.10 (122 days) is the validation set, and the data from 2013 to 2015 (1095 days) is 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) is reserved for another comparison.

Results of another traditional regression model, i.e. support vector regression (SVR), for SST prediction is given for comparison purpose. We run the experiments under the environment of Intel(R) Core(TM)2 Quad CPU Q9550 @2.83GHz, 6G RAM, Ubuntu 16.10 64 bits operating system, and Python 2.7. The proposed network is implemented by TensorFlow 0.11 [13]. SVR is implemented by Scikit-learn [14].

The performance evaluation of SST prediction is a fundamental issue. In this letter, we use root of mean squared error (RMSE), one of the most commonly used measurement as
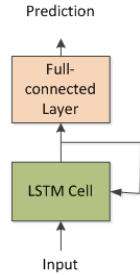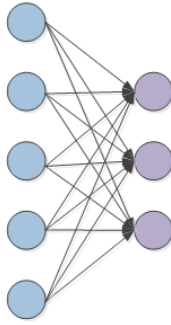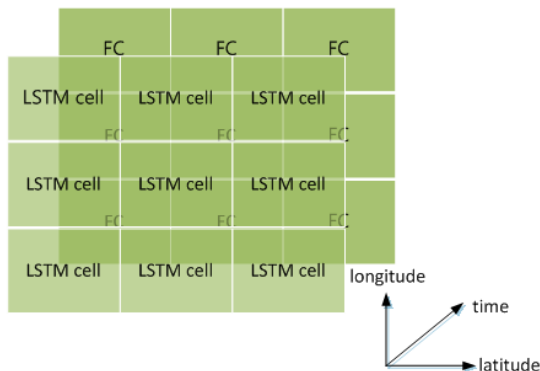
TABLE I: Prediction Results (RMSE&ACC) on Five Positions with Different $units\_r$s.

$units\_r$	metrics	p_1	p_2	p_3	p_4	p_5
3	RMSE	0.5757	0.5907	3.3174	0.8656	5.6695
	ACC	0.9824	0.9825	0.9276	0.9725	0.8593
4	RMSE	0.6026	0.5412	0.8191	0.8156	0.8222
	ACC	0.9820	0.9838	0.9718	0.9737	0.9718
5	RMSE	0.5866	0.5463	0.8445	0.8581	**0.7441**
	ACC	0.9818	0.9834	0.9711	0.9729	**0.9742**
6	RMSE	**0.5649**	**0.5254**	**0.7663**	0.8125	0.8176
	ACC	**0.9829**	**0.9843**	**0.9730**	0.9738	0.9721
7	RMSE	0.5820	0.5302	0.7790	**0.7816**	0.7470
	ACC	0.9825	0.9841	0.9728	**0.9749**	0.9741

the evaluation metric to measure the effectiveness of different methods. In addition, we define a metric to evaluate the prediction accuracy as follows:

$$ACC = \frac{1}{n} \sum_{i=1}^{n} \frac{|prediction - true\ value|}{true\ value} \qquad (5)$$

where n is the prediction length.

RMSE is the smaller the better, while ACC is the opposite. Here RMSE and ACC can be regarded as absolute error and relative error. And for area prediction, we use the area average RMSE and area average ACC.

C. Determination of parameters

We randomly choose 5 positions in Bohai Sea denoted as $p_1, p_2, ..., p_5$ to predict 7 days' SST values with lead time of one month (30 days). Firstly, we fix l_r and l_{fc} as 1, $units\_fc$ as 7, and choose a proper value for $units\_r$ from {3,4,5,6}. Table I shows the results on five positions with different values of $nunits\_r$. The boldface items in the table represent the best performance, i.e. the smallest RMSE and the largest ACC. It can be seen from the results that the most best performance occurs when $units\_r = 6$.

In this experiment, the best performance occurs when $units\_r = 6$ in three positions p_1, p_2 and p_3, while occurs when $units\_r = 5$ at p_5 and $units\_r = 7$ at p_4. But we can see that the difference of RMSE and ACC is not too large. So in the following experiments, we set $units\_r6$ as 6.

Then, we also use the SST sequences from the same five positions to choose a proper value for l_r from {1,2,3}. The other two parameters are set by $unit\_r = 6$, and $l_{fc} = 1$. Table II shows the results on five positions 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 due to the increasing weights numbers with increasing recurrent LSTM layers, which our data is insufficient to train so many weights. Actually, experiences show that the recurrent LSTM layer is not the more the better. And during the experiments we find that more LSTM layers are more likely to get unstable results. So in the following experiments, we set l_r as 1.

Lastly, we still use the SST sequences from the same five positions to choose a proper value for l_{fc} from {1,2}. Though

TABLE II: Prediction Results (RMSE&ACC) on Five Positions with Different l_rs.

l_r	metrics	p_1	p_2	p_3	p_4	p_5
1	RMSE	**0.5649**	**0.5254**	**0.7663**	**0.8125**	**0.8176**
	ACC	**0.9829**	**0.9843**	**0.9730**	**0.9738**	**0.9721**
2	RMSE	3.0357	0.5773	3.3091	0.8289	4.0466
	ACC	0.9442	0.9826	0.9296	0.9736	0.9177
3	RMSE	3.0371	0.5711	0.7991	0.8442	4.0451
	ACC	0.9443	0.9832	0.9721	0.9730	0.9163

TABLE III: Prediction Results (RMSE&ACC) on Five Positions with Different ks.

l_{fc}	metrics	p_1	p_2	p_3	p_4	p_5
1[7]	RMSE	0.5649	**0.5254**	**0.7663**	**0.8125**	**0.7376**
	ACC	0.9829	**0.9843**	**0.9730**	**0.9738**	**0.9789**
2[7,7]	RMSE	**0.5533**	0.5266	0.7805	3.1091	6.9153
	ACC	**0.9832**	0.9842	0.9730	0.9357	0.8044
2[10,7]	RMSE	0.5794	0.5298	3.3422	6.0412	5.6626
	ACC	0.9823	0.9840	0.9265	0.8235	0.8617
2[15,7]	RMSE	3.0349	2.6857	0.7856	3.1001	0.7430
	ACC	0.9454	0.9510	0.9645	0.9373	0.9742

the number of the hidden units of the full-connected layer is tricky. 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 achieve the most best performance when $l_{fc} = 1$. The reason may be the same: more layers means more weights to be trained and more computation it needs. So in the following experiments, we set l_{fc} as 1, and the number of its hidden units is set the same as the prediction length.

D. Results and Analysis

We use Bohai SST data set to do this experiment, and compare the proposed method to a classical regression methods SVR [15]. The setting is as follows. For LSTM network, we set $k = 10, 15, 30, 120\ for\ l = 1, 3, 7, 30\ represently, and\ l_r = 1, l_{fc} = 1$. For SVR, we use the RBF kernel and set the kernel width $\sigma = 1.6$ which is chosen by cross validation on validation set.

Table IV shows the results. The boldface items in the table represent the best performance, i.e. the smallest area average RMSE and the largest area average ACC. It can be seen from the results that the LSTM network achieve the best prediction performance. And Fig.5 shows the SST prediction at one position using two different methods. In order to see the results clearly, we only show the prediction results for one year. Green line represents the true value. Red line represents the prediction results of the LSTM network, and blue line represents the prediction results of SVR with RBF kernel.

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 call the model trained above *original model*, and

TABLE IV: Prediction Results (Area Average RMSE & ACC) on Bohai Sea Data Set.

Methods	Metrics	Prediction Length			
		1 (one day)	3 (three days)	7 (one week)	30 (one month)
SVR	RMSE	0.3998	0.6158	0.8388	1.2477
	ACC	0.9872	0.9802	0.9728	0.9593
LSTM network	RMSE	**0.0767**	**0.1775**	**0.6540**	**1.1363**
	ACC	**0.9923**	**0.9878**	**0.9795**	**0.9690**

TABLE V: Prediction Results(Area Average RMSE & ACC) on Bohai Sea Data Set in 2016.

Model	Metrics	Prediction of 2016			
		1	3	7	30
original	RMSE	0.1346	0.2145	0.6891	1.1521
	ACC	0.9812	0.9887	0.9711	0.9606
updated	RMSE	**0.0899**	**0.1843**	**0.5825**	**1.0123**
	ACC	**0.9905**	**0.9804**	**0.9798**	**0.9701**

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 V shows the results of SST prediction for 2016 using these two different models. The updated model performs the best as expected.

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 going on. While other regression models like SVR don't 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, we formulate the prediction of SST as a time series regression problem, and propose a LSTM based network to model the temporal relationship of SST to predict the future value. This is the first time, to our knowledge, to use recurrent neural network to solve the prediction problem of SST. The proposed network utilizes LSTM layer to model the time series data, and full-connected layer to map the output of LSTM layer to a final prediction. We explore the

optimal setting of this architecture by experiments and report the prediction performance of coastal seas of China to confirm the effectiveness of the proposed method. In addition, we also show the online update characteristics of the proposed method.

And furthermore, the proposed network is independent of the resolution of data. If a high resolution prediction is wanted, all that is needed is to provide a high resolution training data to the network. Once we get the predicted SST values in the future, it can be used in many applications including ocean front prediction, abnormal event prediction, etc.

REFERENCES

[1] K. Patil, M. C. Deo, and M. Ravichandran, "Prediction of sea surface temperature by combining numerical and neural techniques," *Journal of Atmospheric & Oceanic Technology*, 2016.

[2] J. Kug, I. Kang, J. Lee, and J. Jhun, "A statistical approach to indian ocean sea surface temperature prediction using a dynamical enso prediction," *Geophysical Research Letters*, vol. 31, no. 9, pp. 399–420, 2004.

[3] I. D. Lins, M. Moura, M. Silva, E. L. Droguett, D. Veleda, M. Araujo, and C. M. C. Jacinto, "Sea surface temperature prediction via support vector machines combined with particle swarm optimization," 2010.

[4] Wikipedia, "Recurrent neural network," https://en.wikipedia.org/wiki/Recurrent_neural_network.

[5] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–44, 2015.

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

[7] A. Graves, "Generating sequences with recurrent neural networks," *Computer Science*, 2013.

[8] N. Kalchbrenner, I. Danihelka, and A. Graves, "Grid long short-term memory," *Computer Science*, 2015.

[9] N. ESRL, "Noaa oi sst v2 high resolution dataset," http://www.esrl.noaa.gov/psd/data/gridded/data.noaa.oisst.v2.highres.html.

[10] Wikipedia, "Bohai sea," https://en.wikipedia.org/wiki/Bohai_Sea.

[11] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. 7, pp. 257–269, 2010.

[12] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, A. Ranzato, A. Senior, and P. Tucker, "Large scale distributed deep networks," *Advances in Neural Information Processing Systems*, pp. 1232–1240, 2012.

[13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[15] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines." *Advances in Neural Information Processing Systems*, vol. 28, no. 7, pp. 779–784, 1996.
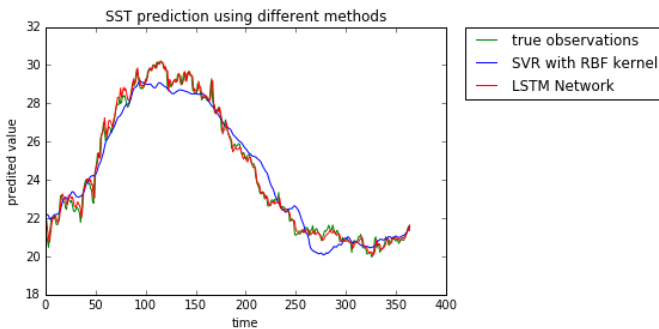

Fig. 5: SST Onr Month Prediction at One Position Using Different Methods