Applications of Deep Learning to Predict Ocean-Atmospheric Characteristics

Atefeh Pourkarimian[1] Mahdi Mohammad Mahdizadeh* Mohammad Reza Khalilabadi[2]

Abstract

Oceanographic observations and models are imperfect, and therefore our simulations of the ocean are not completely realistic. Direct measurements of oceanic processes and properties are limited by sampling rates, while ocean models are limited by finite resolution, high viscosity and diffusion coefficients are needed in solving equations. This study instead evaluated deep learning methods, which focused on data as opposed to equations. There are used particular types of deep learning algorithms and hybrid models consists of artificial neural network (ANN), convolution neural network (CNN), long short-term memory network (LSTM) and etc., to make more accurate the prediction of ocean-atmospheric characteristics include; sea surface wind, sea surface temperature (SST), sea surface salinity (SSS) and sea surface height (SSH). Ocean time series data available in the databases preprocessed to achieve an appropriate pattern and predict factors for short-term in the oceanic area. The total framework of the simulation includes six main stages. At first, the data have been collected and prepared, and then trained the model. Moreover, the proposed hybrid model implemented and validated to predict the studied parameter for short-term (a period of several hours to several days) in several geographical points in a local Sea. Finally, the model performance evaluated and compared the accuracy through MAE, MSE and RMSE criteria. Results show high accuracy for predicted sea surface temperature and salinity in all selected points.

Keywords: Deep Learning, Prediction, Sea Surface Temperature, CNN, BLSTM, Ocean Characteristics

1 Introduction

In recent years, many researches have been focused on the application of various deep learning techniques, including combined methods in atmospheric and oceanic sciences, to predict the key parameters of ocean water, and now these methods are being developed and will be replaced with many traditional methods in the near future [14]. Therefore, studying and investigating new techniques based on deep learning can produce new knowledge and insight in this direction. The main goal of this research is to provide and develop a quick, low-cost and practical solution to increase the accuracy of predicting marine parameters through deep learning. In other words, we want to validate a new deep learning method for a more accurate and practical prediction of physical characteristics of the sea water. More accurate marine forecasting methods can directly and indirectly have a significant impact on the exploration, exploitation and extraction of oil and gas resources from the sea, fisheries, weather and

* Department of Atmospheric & Oceanographic Science, University of Hormozgan, **mehdizadeh@hormozgan.ac.ir**
[1] Department of Atmospheric & Oceanographic Science, University of Hormozgan, **a.pourkarimaian.phd@hormozgan.ac.ir**
[2] Faculty of Naval Aviation, Malek Ashtar University of Technology, **khalilabadi@mut.ac.ir**

climate forecasts, the assessment and protection of marine environment and ecosystems, as well as the better understanding of sea currents and ocean circulations. Both data from observations and ocean models lack information at small and fast scales. Moreover, methods are needed to extract information, extrapolate, or upscale existing oceanographic data sets, to account for or represent unresolved physical processes [1]. Previous numerical and statistical models often do not have the necessary accuracy, in which other factors should also be considered. For example, to predict the wind speed using numerical or general circulation models (GCMs), other meteorological data such as atmospheric pressure, temperature, humidity, dew point, etc. are needed, or to estimate the sea surface level, some other factors such as evaporation, precipitation, and the in and out flow rivers should also be considered [6]. So, we need some other methods to make our ocean data more realistic and complete to achieve a more accurate and closer understanding of the real oceans. Scientists have traditionally approached this problem in a pen-and-paper style, considering physical theories and mechanisms [1]. However, in the methods based on deep learning, without the need for any other additional parameters, it is possible to provide predictions for the future with appropriate accuracy only by training the data of the past time [6]. This type of neural network works well even if ocean data are limited to a particular region and has recently been used in various research fields [3], [6], [9], [10], [11], [12], [13], that indicated an increase in prediction accuracy. The purpose of hybrid Deep Learning models is to obtain an optimal forecasting performance. By using the combined method, it is possible to maximize the information, integrate the information of the models, and improve accuracy of the predictions [2]. In this paper, we intend to adopt a suitable approach to predict each of the atmospheric-oceanic parameters by comparing and reviewing existing deep learning models.

2 Methods of investigation

Data collecting is the first step in all deep learning methods. There are some typical databases used in the previous studies include:

Optimized and Interpolated Sea Surface Temperature (OISST) data provided by the US National Oceanic and Atmospheric Administration (NOAA) includes the daily, weekly and monthly mean SST datasets, which is produced by an advanced radiometer with very high resolution. The daily OISST database covers global oceans from 75.89°S to 89.75°N and from 0.25°E to 359.25°E. This data has a spatial resolution of 0.25° x 0.25°. OISST weekly and monthly data cover from 89.5°S to 89.5°N and from 0.5°E to 359.5°E with a spatial resolution of 1° × 1°.

UK Hadley Center for Sea Surface Temperature and Sea Ice Data, which provides monthly datasets with 1° × 1° spatial resolution also used. This center analyzes and replicates the observations obtained from the global communication system.

Remote Sensing Systems (RSS) database which provide global and daily SSS data using the SMAP satellite by the average of eight days of salinity with a spatial resolution of 0.25° × 0.25° (with an effective spatial resolution of approximately 40 km). The SMAP satellite was launched by the National Aeronautics and Space Administration (NASA), which is capable of simultaneously monitoring soil moisture and SSS globally.

The Archiving, Validation and Interpretation of Satellite Oceanographic data (AVISO) captures and integrates sea surface height (SSH) data from multiple satellites. These data have a spatial resolution of 0.25° × 0.25° with a monthly time resolution.

These reanalyzed data are presented by the Copernicus Marine Environment Monitoring Service (CMEMS) for quality control.

ECMWF Reanalysis v5 (ERA5) is the fifth generation ECMWF atmospheric reanalysis of the global climate covering the period from January 1940 to present. ERA5 is produced by the

Copernicus Climate Change Service (C3S) at ECMWF. ERA5 provides hourly estimates of wind. The data cover the Earth on a 31km grid and resolve the atmosphere using 137 levels from the surface up to a height of 80km. ERA5 includes information about uncertainties for all variables at reduced spatial and temporal resolutions.

At the second step, collected data should be preprocessed, which follows with two phases; data cleansing and standardization. In order to store, read, writing and standardizing the data which is consist of several dimensions (time, longitude and latitude) and several variables (SST, SSS, SSH and wind), there is required a reliable and flexible data structure named Netcdf in order to store the multidimensional data. Each of the datasets belongs to a geographical area and has a significant resolution which the model is grided accordingly. A spherical coordinate system also used, which includes longitude and latitude.

Before training models, datasets must be pre-processed. The input data contain more than one variable with different scales and units. Differences in the scale of the input variables may increase the difficulty of the prediction and make the model unstable and result in adverse performance during the training phase. Therefore, standardization is required to make the same scales of all features, and to be avoided any bias in the model. The mathematical relationship of standardization shown as:

(1) $$Z_{i,j} = \frac{x_{i,j}-\mu_j}{\sigma_j}$$

$$i = 1, \dots, n \qquad j = 1, \dots, m$$

In the above relationship, n is the number of data and m is the number of features or variables, which shows that standardization is done for each feature separately and for all data. The mean value of each feature μ and the standard deviation of each feature σ are calculated from the following relationships, respectively:

(2) $$\mu = \frac{\sum_{i=1}^{N} x_i}{N}$$

(3) $$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i-\mu)^2}{N}}$$

At the third step, we are represented the structures and algorithms of deep learning models. Models should be writing with an appropriate Programming Language and implement. Python programming language by using open-source tools and libraries (tKinter, Numpy, SKLearn, Pickle, Shutil, Keras, TensorFlow, Matplotlib) can access Netcdf data. Netcdf data has a very complex structure and is not suitable for data mining and machine learning. However, Python has made it possible to convert this type of data into the standard format of data engineering called Data Frame. In the following, the different types of deep learning algorithms are presented and described:

2.1 Convolution Neural Network (CNN)

CNN is a subset of Deep Feed-Forward Artificial Neural Networks [7]. The CNN models were originally developed for image classification. These models accept two-dimensional input image with color channels to learn its features. Such models are deep learning methods and have achieved tremendous success in the past. A one-dimensional version of CNN is termed as 1D CNN. The 1D CNN is mainly applied to one-dimensional sequence of data. It extracts important features from the input sequence data and maps the internal feature of the sequence. The 1D CNN has been successfully applied for time series and fixed-length signal data analysis such as audio recordings and natural language processing. Figure 1 shows the CNN model architecture. The CNN consist of a 1D convolutional layer, a pooling layer, a flatten layer, and an output layer. The input signal can be either multivariate or univariate time series. The width of the time series depends on the number of features *K* and the length *N* of the series. The convolutional filters have the same width as the width of the time series but their lengths may be different. The filters are designed to move in one direction while performing a convolutive operation from the starting point of the time series to its endpoint. The convolutional layer consists of new filtered times vectors whose numbers depend on the number of convolution kernels. This layer also captures the features of the initial time series. The next stage involves the pooling of each time series vector of the convolutional layer to form new vectors. The layer responsible for pooling is termed as the pooling layer. The vectors from the pooling layer are passed to the flattened layer or fully connected layer [6]. In the present case, the output of the flattened layer is passed to the LSTM neural network.

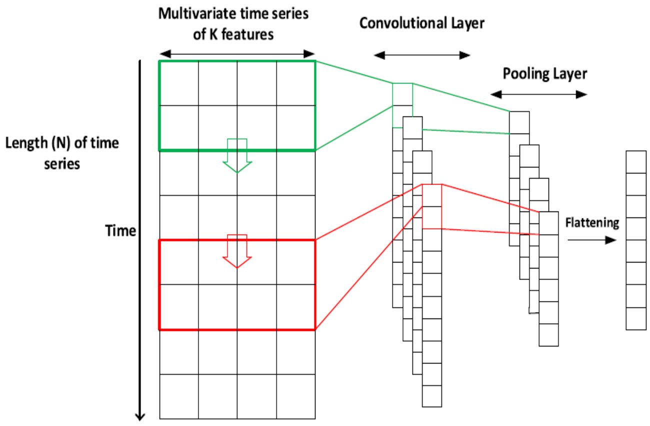

Figure 1: The CNN model architecture [6].

2.2 Long Short-Term Memory (LSTM)

LSTM is a subclass of RNN model which is formed by adding a memory cell into the hidden layer to control the memory information of the time series [6]. It consists of three different control gates namely, forget, input, and output. The state of the memory cell of the LSTM is controlled by two of these gates. The forget gate indicates how much memory of the last moment can be saved while the input gate determines how much input of the current moment can be saved and also controls the fusion of information and stimulus. The output gate is mainly used to control the amount of information that is sent for cell status. The transmitted information passes through the controllable gates to different cells in the hidden layer. This enables the control of the memory and forgetting extent of the prior and current information. In contrast to the RNN, the LSTM has the long-term memory function and does not have the problem of gradient disappearance. Figure 2 shows the structure of the LSTM network.

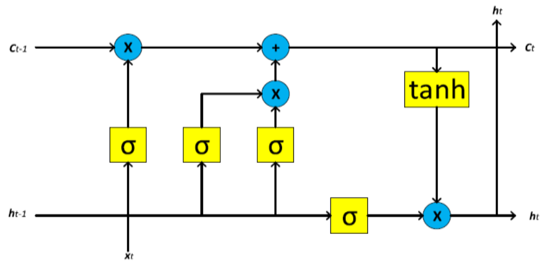

Figure 2: LSTM diagram [6].

σ is the sigmoid function shown in Equation (4)-(6) and has a value between zero and one, where 0 indicates that nothing passes while 1 means everything passes. The hyperbolic tangent function is used to overcome the problem of gradient disappearance. The subscripts i, f and o represent the input, forgetting, and output respectively, and the subscript t represents the time step- index. The equations are given as follows;

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{4}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{5}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{6}$$

$$\tilde{c}_t = \sigma_h(W_c x_t + U_c h_{t-1} + b_c) \tag{7}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{8}$$

$$h_t = o_t \odot \sigma_h(c_t) \tag{9}$$

where W_f, W_i, W_o and W_c are matrices representing the weights of the forgetting gate, input gate, output gate and the memory cell; respectively. U_f, U_i, U_o, U_c are the matrices representing the weights of the recurrent connections of the forgetting gate, input gate, output gate and the memory cell; respectively. x_t denotes the input vector to the LSTM network at a time step t, f_t denotes the forget gate's activation vector, i_t represents the input gate's activation vector, and o_t is the output gate's activation vector, ⊙ represents the element wise multiplication. $\tilde{c}_t$ and c_t represent the cell input activation vector and cell state activation vector. Here, the b_f, b_i, b_o and b_c represent the forgetting gate bias vector, input gate bias vector, output gate bias vector, and memory cell bias vector; respectively. The sigmoid function $\sigma(x)$ and the hyperbolic tangent function $\sigma_h(x)$ are defined as:

(10) $$\sigma(x) = \frac{1}{1+e^{-x}}$$

(11) $$\sigma_h(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$

Bidirectional Long Short-Term Memory (BLSTM) achieve this by presenting the input data forwards and backwards to two separate hidden layers, both of which are connected to the same output layer (see Figure 3).

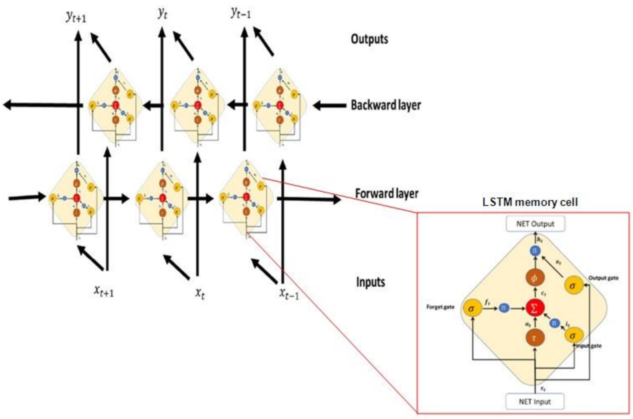

Figure 3: Bidirectional LSTM layer [7].

2.3 Hybrid 1D CNN-BLSTM Model

CNN-BLSTM is presented for the time series prediction as shown in Figure 4. The CNN will extract important high-level features from the input time series. These features will be sent as input to the BLSTM to support prediction after pooling and flattening. The convolution layer will be initialized with 32 different kernels of the same size (3 times 3) and the output of this layer will be passed to the ReLU activation function. To reduce the sensitivity of feature map to location, the max pooling will be employed to select the maximum value and hence reducing the size of feature maps. An BLSTM of 32 output units will be used. The network output will be obtained from the dense output layer. It is noteworthy to know that the output of the network could be increased to multiple feature prediction.

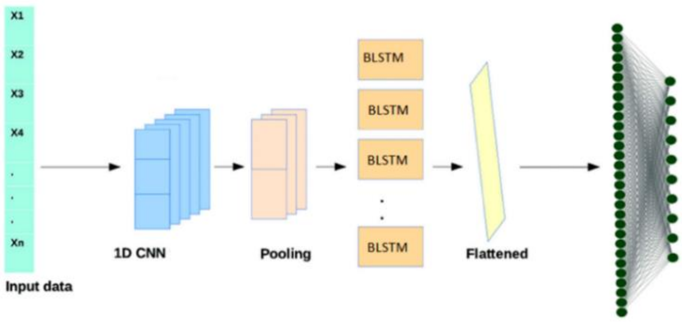

Figure 4: The hybrid CNN-BLSTM model architecture [5].

2.4 Recurrent Neural Networks (RNN)

Traditional neural networks usually assume that all inputs (or outputs) are independent of each other. However, in the process of practical operation, there is a dependency between the current state of each node and the previous steps, and this is the basic assumption of expanded RNN. The signal feedback structure of the recurrent neural networks (RNN) adopts the output state of the network at the time of K associated with the historical signal before the time of K, in order for it to have dynamic characteristics and a memory capability. However, RNNs are challenged by the vanishing gradient problem, where the gradient decreases over time. Moreover, the RNN may also suffer from the gradient explosion problem. Although many techniques have been developed to address this issue, it remains difficult to obtain long-term memory.

2.5 Gated Recurrent Unit (GRU)

Due to its complex internal structure, the training of the LSTM network is the very time-consuming and the LSTM exhibits a poor real-time capability. With the rapid growth in demand for speech-to-text applications, computing resources are currently not even keeping up with its needs. To solve this problem, Gated Recurrent Unit (GRU) network model was proposed on the basis of the original LSTM model. The forget gate and the input gate are combined into a single update gate, and the cell state, the hidden state and other changes are also mixed. The GRU neural network has been successfully applied to sequential or temporal data. The GRU has a simpler structure than the LSTM; nevertheless, its performance is comparable with the LSTM. The GRU even outperforms LSTM but has a lower complexity and faster convergence. However, the GRU has a serial structure, which makes parallel computation hard to implement.

2.6 Simple Recurrent Unit (SRU)

The common feature of LSTM and GRU is that the calculation of the gate of each time step depends on the output of the previous time step, which leads to a high serial dependence of the network. Also, it is difficult to speed up the calculation by parallel calculation. To solve this problem, the Simple Recurrent Unit (SRU) network was proposed. The main design feature of the SRU is that the gate calculation depends only on the current input cycle. In this way, only the point-by-point matrix multiplication of the model depends on the previous time step. Thus, the network can be configured in parallel. In addition, the SRU also reduces the number of gates, and the design only features the forget gate and the reset gate. In this way, the calculation efficiency of SRU neural network is higher than that of LSTM and GRU.

At the fourth step, dataset separated into two parts: the training and the test data. In order to train each model, a separation rate of 80% is used for the test and training data. This means that 20% of the data is used for testing and the remaining 80% for training. Then, the model keeps the data as a series of points in its memory, fit a curve around these points and learns the pattern of each variable fluctuations at these particular points. During the training, model will be able to produce output according to the inputs. This procedure will be occurring in the model by auto-regression technique. There is a correlation between the past, present and the future time in the dataset. There are several windows through the auto-regression, called Sliding Window, which can estimate data for the future by looking at the past time data. The auto-regression approach is shown in the figure 5.

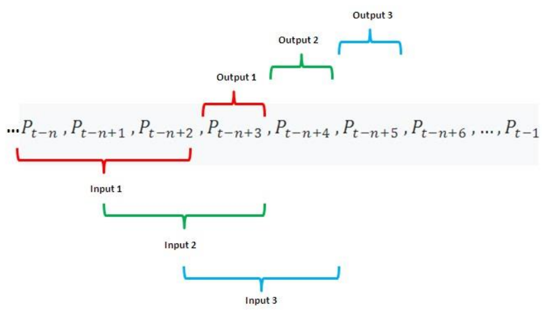

Figure 5: Auto-regression approach in training.

At the fifth step, model have implemented for a short period of time, and validated for some geographical selected points in the oceanic area (Figure 6).

Figure 6: Oceans of the earth.

At the final step, several performance indices will be used to comprehensively evaluate the forecasting capabilities of the model. The performance indices include MAE, MSE, and RMSE and R^2. R-Squared (R^2) is an important statistical measure of fit which indicates how much variation of a dependent variable is explained by the independent variable(s) in a regression model. It ranges from 0 to 1, where the best fit closes to 1. The mathematical expression of the aforementioned indices is given as follows:

(12) $$MSE = 1/N \sum_{n=1}^{N} (y'_n - y_n)^2$$

(13) $$MAE = 1/N \sum_{n=1}^{N} |y'_n - y_n|$$

(14) $$RMSE = \sqrt{1/N \sum_{n=1}^{N} (y'_n - y_n)}$$

(15) $$R^2 = 1 - \frac{\sum_{n=1}^{N} \frac{\left(y'^{(n)} - y^{(n)}\right)^2}{N}}{\sum_{n=1}^{N} \frac{\left(y^{(n)} - \bar{y}\right)^2}{N}}$$

In above relationships, N is the number of test data, y is the output of real data, and y′ is the output of the model. After determining the error percentage, the research questions will be answered and the research hypotheses will be finally rejected or confirmed according to the obtained results.

3 Results & Discussion

In this section, we evaluate and review the results obtained from the different deep learning methods.

In [4], the authors propose DeepOcean, a deep learning framework for spatio-temporal ocean sensing data prediction, which consists of a generative module and a prediction module. They implement the generative module with a multi-layer perceptron (MLP) to capture the spatial dependencies and construct high-resolution data based on sparse observations. The prediction module is implemented with their proposed Multivariate Convolutional LSTM (MVC-LSTM) neural network, which captures both the spatio-temporal dependencies and the interactions of different oceanographic features for prediction. They evaluate the effectiveness of DeepOcean with Argo data, where the proposed framework outperforms fifteen state-of-art baselines in terms of accuracy. Table 1 and 2 show the detail evaluation results.

Table 1: Evaluations of different methods in predicting SST [4].

Method	MAE	RMSE	R^2
RR	0.3073	0.3686	0.8642
KNN	0.0532	0.0883	0.9922
MLP	0.0511	0.0825	0.9932

Table 2: Evaluations of different methods in predicting SSS [4].

Method	MAE	RMSE	R^2
RR	0.6355	0.8533	0.2738
KNN	0.2082	0.3312	0.8906
MLP	0.1577	0.2394	0.9429

They display the evaluations of these methods and the time steps they need in Fig. 7, respectively. They choose different time step lengths for different methods for comparison because different model attains its best performance with the lowest RMSE at different time step length.

In [8], the authors have developed an optimized Simple Recurrent Unit (SRU) deep network for the short- to medium-term prediction of the SSHA using Archiving Validation and International of Satellites Oceanographic (AVISO) data. Detailed experiments were carried out in the Bohai Sea to evaluate the proposed model and it was demonstrated that the proposed framework (1) outperformed significantly the current deep learning methods such as the BP (Backpropagation), the RNN (Recurrent Neural Network), the LSTM (Long Short-term Memory), and the GRU (Gated Recurrent Unit) algorithms for 1, 5, 20, and 300-day prediction;

(2) can predict the short-term trend in the SSHA (for the next day or 2 days) in real time; and (3) achieves medium-term prediction in seconds for the next 5–20 days and shows great potential for applications requiring medium- to long-term predictions (Table 3 to 5).

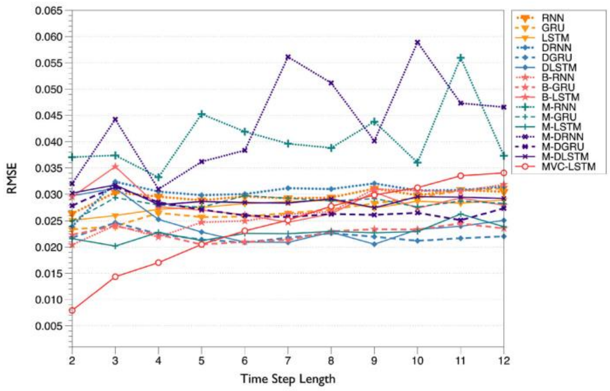

Figure 7: Comparison of MVC-LSTM and other baselines with different time step length using RMSE. The smaller the better [4].

Table 3: The experimental results of 1 and 5-day prediction of SSH [8].

Model	1-day			5-day		
	R^2	RMSE (cm)	Time (s)	R^2	RMSE (cm)	Time (s)
BP	0.903	2.60	12	0.874	4.6	18
LSTM	0.989	1.05	9	0.971	1.13	10
GRU	0.987	1.09	6	0.977	1.03	8
SRU	0.99	1.03	1	0.987	1.01	2

Table 4: The experimental results of 20-day prediction of SSH [8].

Model	R^2	RMSE (cm)	Time (s)
BP	0.743	6.86	24
LSTM	0.968	1.17	18
GRU	0.972	1.12	12
SRU	0.974	1.09	9
SRU*	0.994	0.89	4

GPU-based parallel SRU algorithm

Table 5: The experimental results of 300-day prediction of SSH [8].

Model	R^2	RMSE (cm)	Time (s)
BP	0	55.48	500
LSTM	0.984	1.37	380
GRU	0.971	1.67	290
SRU	0.982	1.35	268
SRU*	0.991	1.05	210

GPU-based parallel SRU algorithm

In [5], the authors proposed a deep-learning-based wind speed forecasting model based on CNNs and BLSTM. The simulation results illustrate the accurate and reliable performance of the proposed method. Also, it is shown that the performance of the model in forecasting the U characteristics of the wind is relatively better than V characteristics (Figure 8 & 9). Table 6 shows the evaluation metrics and the training time for the prediction methods.

Table 6: Evaluation metrics for prediction of wind [5].

	LSTM	ANN	CNN_BLSTM
MSE	4.9176	6.912074	3.787418
RMSE	2.979293	3.21632	1.946129
MAE	2.526305	3.728342	1.473448
Training time (minutes)	27.5	21	38.5

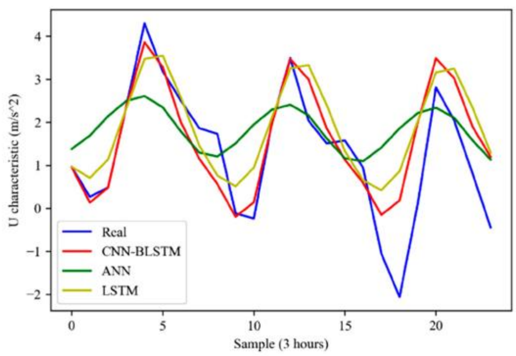

Figure 8: The regression diagram of the U values for a period of three days [5].

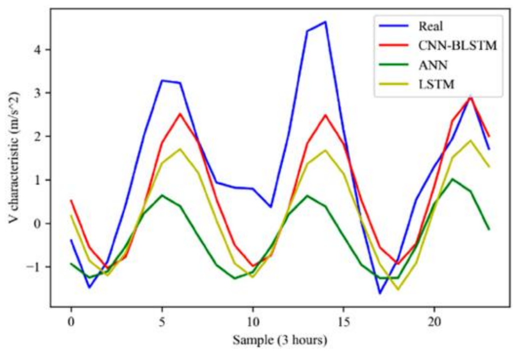

Figure 9. The regression diagram of the V values for a period of three days [5].

4 Conclusion

From the presented results and evaluations, it can be concluded that the deep learning hybrid models have acceptable performance in predicting ocean parameters. It should be noted that due to the periodic pattern and alternative nature of sea surface temperature and salinity, it is very convenient and easier for the models to predict these oceanic parameters relative to the sea surface wind and water level fluctuations, which are very variable and irregular in nature.

References

[1] Bolton, T., & Zanna, L. (2019). Applications of deep learning to ocean data inference and subgrid parameterization. Journal of Advances in Modeling Earth Systems, 11(1), 376-399.

[2] Chang, W. Y. (2014). A literature review of wind forecasting methods. Journal of Power and Energy Engineering, 2(04), 161.

[3] Du, S., Li, T., Yang, Y., &Horng, S. J. (2019). Deep air quality forecasting using hybrid deep learning framework. IEEE Transactions on Knowledge and Data Engineering, 33(6), 2412-2424.

[4] Gou, Yu, et al. "DeepOcean: A general deep learning framework for spatio-temporal ocean sensing data prediction." IEEE access 8 (2020): 79192-79202.

[5] Khalilabadi, Mohammad Reza. "Wind field forecasting using a novel method based on convolutional neural networks and bidirectional LSTM." Ships and Offshore Structures 19.7 (2024): 892-900.

[6] Lawal, A., Rehman, S., Alhems, L. M., &Alam, M. M. (2021). Wind speed prediction using hybrid 1D CNN and BLSTM network. IEEE Access, 9, 156672-156679.

[7] Maalej, R., & Kherallah, M. (2018, November). Convolutional neural network and BLSTM for offline Arabic handwriting recognition. In 2018 International Arab conference on information technology (ACIT) (pp. 1-6). IEEE.

[8] Ning, Pengfei, et al. "Short-to medium-term Sea surface height prediction in the bohai Sea using an optimized simple recurrent unit deep network." Frontiers in Marine Science 8 (2021): 672280.

[9] Passricha, V., & Aggarwal, R. K. (2019). A hybrid of deep CNN and bidirectional LSTM for automatic speech recognition. Journal of Intelligent Systems, 29(1), 1261-1274.

[10] Shao, Y. T., & Chou, K. C. (2020). pLoc_Deep-mAnimal: a novel deep CNN-BLSTM network to predict subcellular localization of animal proteins. Natural Science, 12(5), 281-291.

[11] Shen, Q., Wang, Z., & Sun, Y. (2017). Sentiment analysis of movie reviews based on cnn-blstm. In Intelligence Science I: Second IFIP TC 12 International Conference, ICIS 2017, Shanghai, China, October 25-28, 2017, Proceedings 2 (pp. 164-171). Springer International Publishing.

[12] Xia, T., Song, Y., Zheng, Y., Pan, E., & Xi, L. (2020). An ensemble framework based on convolutional bi-directional LSTM with multiple time windows for remaining useful life estimation. Computers in Industry, 115, 103182.

[13] Xiong, W., Wu, L., Alleva, F., Droppo, J., Huang, X., &Stolcke, A. (2018, April). The Microsoft 2017 conversational speech recognition system. In 2018 IEEE international

conference on acoustics, speech and signal processing (ICASSP) (pp. 5934-5938). IEEE.

[14] Zheng, G., Li, X., Zhang, R. H., & Liu, B. (2020). Purely satellite data–driven deep learning forecast of complicated tropical instability waves. Science advances, 6(29), eaba1482.