Applying Bayesian inference in a hybrid CNN-LSTM model for time series prediction

Thi-Lich Nghiem, Viet-Duc Le, Thi-Lan Le, Pierre Maréchal, Daniel Delahaye, Andrija Vidosavljevic

Applying Bayesian inference in a hybrid CNN-LSTM model for time series prediction

Abstract—**Convolutional neural networks (CNN) and Long short-term memory (LSTM) provide state-of-the-art performance in various tasks. However, these models are faced with overfitting on small data and cannot measure uncertainty, which have a negative effect on their generalization abilities. In addition, the prediction task can face many challenges because of the complex long-term fluctuations, especially in time series datasets. Recently, applying Bayesian inference in deep learning to estimate the uncertainty in the model prediction was introduced. This approach can be highly robust to overfitting and allows to estimate uncertainty. In this paper, we propose a novel approach using Bayesian inference in a hybrid CNN-LSTM model called CNN-Bayes LSTM for time series prediction. The experiments have been conducted on two real time series datasets, namely sunspot and weather datasets. The experimental results show that the proposed CNN-Bayes LSTM model is more effective than other forecasting models in terms of Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as well as for uncertainty quantification.**

Index Terms—**Bayesian inference; time series dataset; uncertainty quantification**

I. Introduction

Time series prediction is a field of research with increasing interest that is broadly used in various applications such as economy, bio-medicine, engineering, astronomy, weather forecast, air traffic management. The purpose of time series prediction is to predict the future state of a dynamic system from the observation of previous states [1]. However, in a significant number of prediction problems, we have to face uncertainty, non-linearity, chaotic behaviors and non-stationarity, which deteriorates the prediction accuracy of the model.

In order to deal with these issues, many approaches have been proposed. They can be generally categorized into two types: the statistical approach and the deep learning approach. Statistical approaches such as SARIMA [2], Prophet [3] can predict time series precisely by exploiting the relationship between the original data and the predicted states while deep learning approaches such as LSTM, Transformer can model data with rich temporal patterns and learn high-level representations of features and associated nonlinear functions without relying on experts to select which of the manually-crafted features to employ [1], [4].

Besides evaluating the performance prediction, quantification of uncertainty is considered as one of the most important aspects of the decision-making process [5]. In order to quantify the model's uncertainty, many researchers use Bayesian inference to estimate the uncertainty in the prediction model from probability distributions. As the result, it can be highly robust to overfitting and easily learn from minor datasets. In the Bayesian framework, the posterior distribution provides all information about the unknown parameters. Bayesian inference with different techniques such as Markov Chain Monte Carlo, Laplace approximation, expectation propagation, variational inference have been used to quantify the uncertainty in time series data prediction such as sunspot dataset [6], [7], weather dataset [8], [9], etc.

In this study, we propose to use Bayesian inference in a hybrid model between CNN and LSTM. We test on two real datasets, namely sunspot and weather datasets. In addition, we also compare the proposed model to the statistical models and deep learning models as well as uncertainty quantification. The main contributions of this paper are summarised as follows:

- We apply a Bayesian inference to update the weight of hyper-parameters in a hybrid prediction method that combines CNN and LSTM. We use 1D convolutional layer of CNN to extract the spatial features and LSTM to extract the temporal features of the sunspot and weather datasets.
- We also compare the prediction performance of proposed model with statistical models (SARIMA and Prophet) and deep leaning models (LSTM, GRU, Transformer, and Informer).

- Finally, we illustrate the way to calculate the model's uncertainty used in time series dataset.

The rest of our paper is structured as follows: Section II provides brief review of relevant works for time series prediction. Section III introduces our proposed model while Section IV describes the experimental results of two studies on sunspots and weather prediction. The conclusions and future works are summarized in Section V.

II. Related works

To improve the models' performance prediction for time series dataset, many researchers introduced several statistical, deep learning which attack uncertain complex time series.

Statistical approaches could predict time series precisely by mapping the relationship among both original data and predicted data. These models include the ARIMA family of methods such as AR, ARMA, ARIMA, Random Walk, SARIMA [2], Prophet [3], etc. While SARIMA is to describe the current value in a time series based on prior observed data by adding three new hyper-parameters to determine the AR, moving average and distinguishing terms as well as an additional parameter for the seasonal interval, Prophet is a more current time series predicting method. Although this approach has some similarities to SARIMA, it models the trend and seasonality of time series by combining more configurable flexibility. In Prophet approach, the trend, seasonality, and holiday are the three main features, and holiday is selected to change predictions.

Deep learning has proven to be extremely effective in computer vision, computer gaming, multimedia, and big data-related challenges. Deep learning approaches are also widely used to model time series data. Because of their capacity to collect temporal information, RNNs have proven useful in forecasting time series [10]. Many researchers used deep learning approaches such as RNN, LSTM, GRU [11], [12], Transformer [13] or CNN models to forecast temporal information in time series dataset. [14] proposed to use recursive Levenberg-Marquardt Bayesian in RNN to forecast electricity spot prices as well as compute the uncertainty of the model. Other researchers used CNN to predict wind power [15], LSTM to predict wind speed [16], weather [8], [9], sunspot [10], [17], [18], or combine CNN and LSTM [19], [20], RNN and LSTM [21] to forecast the output in time series datasets. Recently, In 2021, Zhou proposed a novel approach called Informer to deal with heavy memory when using long input sequences [22]. This approach is an improvement of Transformer approach [13]. The main idea of Informer is to use a ProbSparse technique in selecting only the most crucial queries by using Kullback-Leibler. So it can decrease the time complexity and memory useage.

III. Proposed method

A. Long Short-Term Memory

LSTM network is a advanced version of RNN proposed by Hochreiter in 1997 [23]. It is applied very effectively used due to the capability of learning short and long dependencies. The network's (and so RNN) default behavior is to remember information for a long time. RNNs take the form of a repeating sequence of NN modules. In RNN, these modules have a very simple structure, just a $tanh$ layer. But the issue is that RNN cannot process long-term dependency, LSTM is intended to prevent this problem. LSTMs also have a string structure. Instead of a single NN layer, LSTM has four layers which interact with each other (seen in Figure 1).

The main idea of LSTM is that the cells' state is depicted by the horizontal line (red line) at the top, from C_{t-1} to C_t. The cell state is like a carousel running straight through the whole chain with only a few small linear interactions. It is relatively easy for information to remain unaltered.

LSTMs have the ability to remove or add information to the cell state, which is carefully regulated by structures called gates. The gate is an optional way for information to pass through. They are composed of a layer of $sigmoid$ NN and a point-wise multiplication operator. The output of the $sigmoid$ layer are the number values in [0, 1], which describe the throughput of each component. 0 and 1 values mean "let nothing through" and "let everything through", respectively. An LSTM has three $sigmoid$ gates to protect and control the cell state, including the forget, the input, and the output gates.

Hence this allows long-term memory to be reset and overcome the vanishing and exploding gradient problems.

B. Bayesian inference in a CNN-LSTM model

The proposed model named CNN-Bayes LSTM that is illustrated in Fig. 2 has two main parts: CNN (extract the spatial data) and Bayes LSTM (extract long-term temporal data). After the data preparation, high level spatial features can be extracted by using a CNN layer.

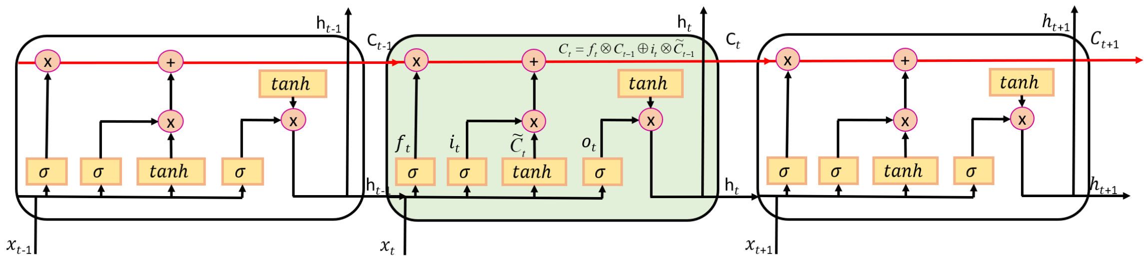


Fig. 1: The LSTM architecture [23].

Then, these extracted low-dimensional features are linked together via a LSTM to extract the temporal features from the data. In this part, we use Bayesian inference to optimize the hyper-parameters and architecture of our model as well as quantify the model's uncertainty on its weights by sampling them from a distribution parameterized by trainable variables on each feed-forward operation. The detailed process in the Bayesian LSTM architecture is illustrated in Figure 3. Finally, a fully-connected layer is added to determine the output.

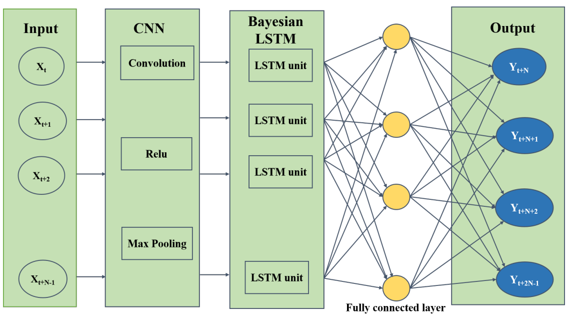


Fig. 2: The CNN-Bayes LSTM framework proposed for time series prediction.

In this paper, we denote X an Y are the input and output of the framework.

- Input: $X = (X_t, X_{t+1}, \ldots, X_{t+N-1})$ where X_t is the observed sample at time t with N is the number of samples.
- Output: $Y = (Y_{t+N}, Y_{t+N+1}, \ldots, Y_{t+2N-1})$ is the predicted values.

For example, we use the temperature of five previous months to predict the future, so $N = 5$. At the first time, $t = 1$, we have $X = (X_1, X_1, \ldots, X_5)$ corresponding to five months from January to May, then the output $Y = (Y_6, Y_7, \ldots, Y_{10})$ corresponding to five months from June to October.

The purpose of Bayesian neural network is rather than having deterministic weights to sample them

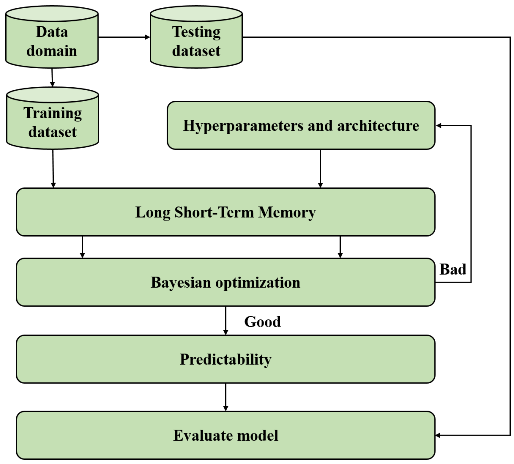


Fig. 3: The Bayesian LSTM flow chart

for a probability distribution and then optimize the distribution parameters. By this approach, it is possible to measure confidence and uncertainty over predictions. In Bayesian LSTM, we can calculate the weights and biases sampling as follows:

- The weight sampled at the i^{th} time on the position N of the layer by the formula:

$$W_{(n)}^{(i)} = \mathcal{N}(0,1) * \log\left(1 + \rho_{(w)}^{(i)}\right) + \mu_{(w)}^{(i)} \quad (1)$$

- The bias sampled at the i^{th} time on the position N of the layer by the formula:

$$b_{(n)}^{(i)} = \mathcal{N}(0,1) * \log\left(1 + \rho_{(b)}^{(i)}\right) + \mu_{(b)}^{(i)} \quad (2)$$

Where ρ and μ are the input feature standard deviation and the input feature mean, respectively.

At the stage, Bayesian optimization calculate the posterior distribution of objective function by using Bayesian inference where the next hyper-parameter combination is selected from this distribution. The previous sampling information is used to find the objective function and the hyper-parameters in order to maximize the target output. If the process is not good, we return to consider the hyper-parameters as well as the architecture in LSTM model. Otherwise, we go to use these weights to predict the future and evaluate the model. In our proposed model, to identify vital LSTM hyper-parameter values, Bayesian optimization is used.

C. *Uncertainty quantification*

Before evaluating forecasting uncertainty, it is necessary to identify the two types of uncertainty (aleatoric and epistemic) and the appropriate solution to decrease them. The first type of uncertainty is epistemic. It refers to model's uncertainty because of the lacking of model's knowledge in features of the input space where there is tiny data such as data sparsity, bias, etc. [24]. It can be reduced by gathering enough data. We can achieve a model's confidence interval by estimating its epistemic uncertainty. The second type of uncertainty is aleatoric. It is essentially a noise inherent in the observations such as input-dependent due to either sensor noise or motion noise which is uniform along the dataset. It cannot be decreased even when more data is collected. We may calculate the prediction interval by estimating the aleatoric and epistemic uncertainty [25], [26]. The confidence interval may be narrower than the prediction interval.

IV. EXPERIMENTAL RESULTS

A. *Dataset*

To evaluate the performance of our proposed model, we test on two real time series datasets, namely sunspot and weather datasets.

1) Sunspot dataset: Sunspot dataset is collected from January 1749 to February 2022 by the research working in the Royal Observatory of Belgium. This data is available at the World Data Center SILSO website [27]. The dataset used in our research includes 3278 samples of averaged total sunspot number per month with the dates and the monthly mean number of sunspots information. It is divided into two sets, including 2294 and 984 samples in training and testing sets, respectively. For the forecasting, the data can be classified into either a fixed time period or a solar cycle. Solar cycles and ordinary years are not distinguished in the dataset. As a result, the dataset only uses the averaged number of sunspots seen in that month.

2) Weather dataset: Weather dataset used in our research includes 1380 samples of the mean temperature values per month in Bangladesh from January 1901 to December 2015. This data is available at Kaggle website [28]. It is divided into two sets, including 965 and 415 samples in training and testing sets, respectively.

Figure 4 illustrates the trend of monthly mean sunspots number and mean temperature from 1901 to 1905. Figure 5 illustrates the monthly mean total sunspot number and temperature in entire two datasets.

B. *Evaluation*

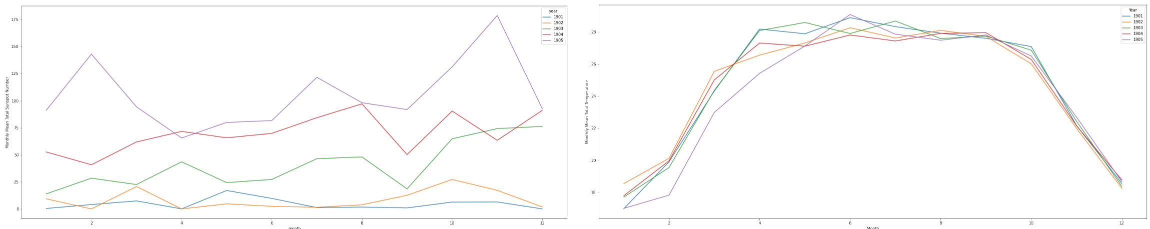
To evaluate the performance of the model's prediction, we use two evaluation metrics in forecasting task, including RMSE and MAE. RMSE is used to measure the magnitude of errors in the prediction and is calculated as quadratic mean of the difference between predicted value and observed value, called prediction error. MAE is a measure of a model's performance in relation to a test set. It captures as the average of the absolute values of individual prediction mistakes across all instants in the test set. RMSE and MAE are defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\widehat{y}_i - y_i)^2};\ \mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\widehat{y}_i - y_i| \tag{3}$$

where $\widehat{y}_i$ and y_i are the observed and predicted values at time step i, n is the length of the sample data.

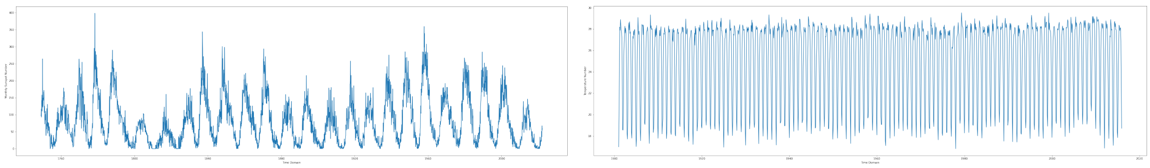
C. *Empirical Results*

Table I shows the results obtained by the proposed method. Furthermore, to show the robustness of the proposed model, we compare the proposed model with others models, namely SARIMA [2], Prophet [3], Transformer [13], Informer [22], LSTM [23], and GRU [11] models. The results show that, our proposed model has outperformed others with 26.10 of RMSE and 18.74 of MAE for sunspot dataset. On weather dataset, the value obtained by the proposed method is 2.23 for RMSE and 1.64 for MAE respectively. It can be clearly seen that, there is a big gap in RMSE values between statistical models and deep learning model. In sunspot dataset, while all used deep learning models have RMSE values under 50 and MAE values from over 22 to under 40, especially Informer model obtained 29.90 at RMSE and 22.25 at MAE, statistical models have over 50 in both RMSE values and MAE values, SARIMA



(a) (b)

Fig. 4: The trend of monthly mean sunspots number (a) and mean temperature (b) from 1901 to 1905 in two datasets.



(a) The monthly sunspot number from 1749 to February 2022 (b) The monthly mean Temperature from 1901 to February 2015

Fig. 5: The trend of monthly mean sunspots number (a) and mean temperature (b) in entire two datasets.

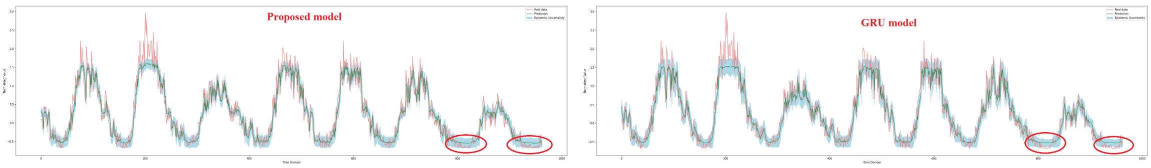
TABLE I: Comparison of the proposed method and the state-of-the-art methods on two datasets. Two best results are in bold.

Forecasting models	Sunspots dataset		Weather dataset	
	RMSE	MAE	RMSE	MAE
SARIMA [2]	54.11	45.51	-	-
Prophet [3]	60.15	56.09	-	-
Transformer [13]	33.99	25.26	**2.10**	**1.43**
Informer [22]	**29.90**	**22.35**	2.32	1.82
LSTM [23]	46.14	39.44	2.32	1.75
GRU [11]	37.14	26.77	4.44	3.43
Proposed model	**26.10**	**18.74**	**2.23**	**1.64**

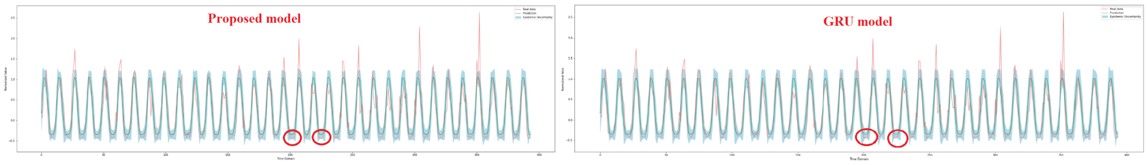
and Prophet models obtained 54.11 versus 33.99 of RMSE and 45.51 versus 56.09 of MAE respectively. Interestingly, on Sunspot dataset, proposed method had an outstanding performance in regarding RMSE and MAE values, it is much better than well-known Informer model (25.95 versus 29.90 in RMSE and 18.61 versus 23.35 respectively). In weather dataset, the RMSE and MAE values are slightly lower than that of Transformer (2.23 versus 2.10 in RMSE and 1.64 versus 1.43 respectively). Our result still is higher than other models such as Informer, LSTM and GRU models. In addition, the proposed model can calculate the epistemic uncertainty. There are some differences between GRU and the proposed models when using Bayesian inference. The Figures 6 are presented the models' epistemic variance estimation on two datasets. It is more interesting differences between these models. We compare GRU and proposed models in three aspects, including the real data, the predicted data and the epistemic uncertainty corresponding to the red line, the green line and the light blue line, respectively. The 95% confidence intervals for the sunspot number and the temperature of two models obtained from numerous predictions are illustrated in this figure. Figure 6 shows that our proposed model captures the variation of the predicted normalized value in the entire two datasets whereas GRU sometimes fails to capture this measure in both datasets (the red circles).

V. Conclusions and Future works

In this paper, we have proposed a novel approach using Bayesian inference in a hybrid CNN-LSTM model called CNN-Bayes LSTM for time series prediction. We evaluated the performance prediction and uncertainty quantification of our proposed model and compared with six models in the literature, including SARIMA, Prophet, Transformer, Informer, LSTM, and GRU in time series dataset forecasting. Experimental results have shown that proposed CNN-Bayes LSTM achieves better performance than existing methods in term of RMSE and MAE values as well as the uncertainty quantification of the model. However, we only used 1D CNN



(a) sunspot data



(b) weather data

Fig. 6: Models' uncertainty quantification in two datasets.

and one factor such as the sunspot number and the temperature. It is interesting idea if we can test on many factors in high dimension dataset (3D or 4D).

REFERENCES

[1] Li, F. Zhang, L. Gao, Y. Liu, and X. Ren, "A novel model for chaotic complex time series with large of data forecasting," *Knowledge-Based Systems*, vol. 222, 2021.

[2] Box and Jenkins, "Time series analysis: Forecasting and control," *Holden-Day Series in Time Series Analysis*, pp. 161–215, 1976.

[3] S. Taylor and B. Letham, "Forecasting at scale," *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018.

[4] Y. Dang, Z. Chen, H. Li, and H. Shu, "A comparative study of non-deep learning, deep learning, and ensemble learning methods for sunspot number prediction," *arXiv*, 2022.

[5] M.Abdar, F. Pourpanah, S. Hussain, and R. D. .etc, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *CoRR*, 2020.

[6] M. Sophie, R. Sachs, C. Ritter, V. Delouille, and L. Laure, "Uncertainty quantification in sunspot counts," *The Astrophysical Journal*, vol. 886, no. 1, pp. 1–14, 2019.

[7] M. Atencia, R. Stoean, and G. Joya, "Uncertainty quantification through dropout in time series prediction by echo state networks," *Mathematics*, vol. 8, no. 8, 2020.

[8] A. Shafin, "Machine learning approach to forecast average weather temperature of bangladesh," *Global Journal of Computer Science and Technology: Neural & Artificial Intelligence*, vol. 19, pp. 39–48, 2019.

[9] T. Siddique, S. Mahmud, A. Keesee, C. Ngwira, and H. Connor, "A survey of uncertainty quantification in machine learning for space weather prediction," *Geosciences*, vol. 12, no. 1, 2022.

[10] R. Chandra, S. Goyal, and R. Gupta, "Evaluation of deep learning models for multi-step ahead time series prediction," *arXiv:2103.14250*, 2021.

[11] K. Cho, B. Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *CoRR*, 2014.

[12] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, 2014.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, 2017.

[14] D. Mirikitani and N. Nikolaev, "Recursive bayesian recurrent neural networks for time-series modeling," *Transactions on Neural Networks*, vol. 21, no. 2, pp. 262–274, 2010.

[15] K. Amarasinghe, D. L. Marino, and M. Manic, "Deep neural networks for energy load forecasting," in *International Symposium on Industrial Electronics*, 2017, pp. 1483–1488.

[16] J. Wang and Y. Li, "Multi-step ahead wind speed prediction based on optimal feature extraction, long short term memory neural network and error correction strategy," *Applied Energy*, vol. 230, pp. 429–443, 2018.

[17] Z. Pala and R. Atici, "Forecasting sunspot time series using deep learning methods," *Solar Physics*, pp. 1–14, 2019.

[18] T. Khan, F. Arafat, U. Mojumdar, A. Rajbongshi, T. Siddiquee, and R. Chakraborty, "A machine learning approach for predicting the sunspot of solar cycle," in *International Conference on Computing, Communication and Networking Technologies*, 2020, pp. 1–4.

[19] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *CoRR*, 2015.

[20] C.-J. Huang and P.-H. Kuo, "A deep CNN-LSTM model for particulate matter (PM2.5) forecasting in smart cities," *Sensors*, vol. 18, no. 7, 2018.

[21] Y. Sudriani, I. Ridwansyah, and H. A. Rustini, "Long short term memory (LSTM) recurrent neural network (RNN) for discharge level prediction and forecast in cimandiri river, indonesia," *IOP Conference Series: Earth and Environmental Science*, vol. 299, 2019.

[22] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," *CoRR*, 2021.

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

[24] Y. Dar, V. Muthukumar, and R. G. Baraniuk, "A farewell to the bias-variance tradeoff? an overview of the theory of over-parameterized machine learning," *arxiv*, 2021.

[25] B. Kappen and S. Gielen, "Practical confidence and prediction intervals for prediction tasks," *Prog. Neural Process*, vol. 8, pp. 128–135, 1997.

[26] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *CoRR*, 2017.

[27] SILSO World Data Center, "The international sunspot number," *International Sunspot Number Monthly Bulletin and online catalogue*, 1749–2022.

[28] www.kaggle.com/yakinrubaiat/bangladeshweather-dataset.