Forecasting Precipitable Water Vapor Using LSTMs

Mayank Jain*, Shilpa Manandhar†, Yee Hui Lee†, Stefan Winkler‡, and Soumyabrata Dev*§

* School of Computer Science, University College Dublin, Ireland

† School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore

‡ School of Computing, National University of Singapore (NUS)

§ ADAPT SFI Research Centre, Dublin, Ireland

***Abstract*—Long-Short-Term-Memory (LSTM) have been extensively used for time series forecasting in recent years due to their ability of learning patterns over both long and short periods of time. In this paper, this ability has been exploited to learn the pattern of Global Positioning System (GPS)-based Precipitable Water Vapor (PWV) measurements over a period of 4 hours. The trained model reported a root mean square error (RMSE) of 0.098 millimeter for a forecasting of 5 minutes in the future, and outperforms naive approach for a lead-time of upto 40 minutes. These RMSE values were computed over the whole test set comprising of more than 1500 hours of recorded data.**

I. Introduction

Over the recent years, GPS (Global Positioning System)-based PWV (Precipitable Water Vapor) values have proved very helpful in determining/forecasting rainfall events [1], [2]. However, this shifts the focus of forecasting from rainfall events to the GPS-based PWV values. On a separate note, Long Short-Term Memory (LSTM) have shown their potential in catering the problems of time series forecasting [3]. Utilizing this potential, an LSTM-based Deep Neural Network (DNN) has been designed and trained in this paper [1] to successfully forecast GPS-based PWV values with high accuracy.

II. GPS-based PWV Measurements

A. PWV Dataset and Pre-processing

The PWV values are computed from the GPS measurements. The GPS signals are usually affected by two primary delays in the atmosphere – Zenith Hydrostatic Delay (ZHD) and Zenith Wet Delay (ZWD). The ZWD delay occurs owing to the water vapor content in the atmosphere. We compute PWV fom the ZWD delays as follows:

$$PWV = PI \cdot ZWD \tag{1}$$

$$\begin{aligned} PI = [&-\mathrm{sgn}(L_a) \cdot 1.7 \cdot 10^{-5} |L_a|^{h_{fac}} - 0.0001] \\ &\cdot \cos \frac{2\pi(DoY - 28)}{365.25} + 0.165 - 1.7 \cdot 10^{-5} |L_a|^{1.65} + f, \end{aligned} \tag{2}$$

where L_a refers to the latitude, DoY is day-of-year, the value of h_{fac} is 1.48 for stations in northern hemisphere and 1.25 for that belonging to southern hemisphere. We compute $f = -2.38 \cdot 10^{-6} H$, where H is the station height, and the ZWD values are processed for a tropical IGS GPS station, ID: NTUS (1.30°N, 103.68°E).

A windowed dataset is required for training the LSTM-based deep neural network for time-series. In this case, each window is a continuous slice of PWV measurements for 4 hours straight (i.e. 48 consecutive readings). The output label is the predicted value or the next consecutive reading in the dataset (i.e. 49^{th} consecutive reading following the values considered for the corresponding input window). Presence of multiple gaps (missing values in the raw data) has also been considered while pre-processing the dataset. This ultimately led to 90011 windows of consecutive readings. In other words, this accounted for more than 7500 hours of PWV measurement data. The first 80% of this pre-processed dataset was used for training the network, while the remaining was used for testing and reporting results.

B. Forecasting Methodology

An LSTM-based deep neural network (see Fig. 1) has been trained for the task of predicting for a lead-time of 5 minutes (i.e. immediate next step in series) given the past data of consecutive 4 hours. Similar to the Recurrent Neural Network Language Model (RNNLM) [4], the trained network is used to forecast PWV values ahead in future.

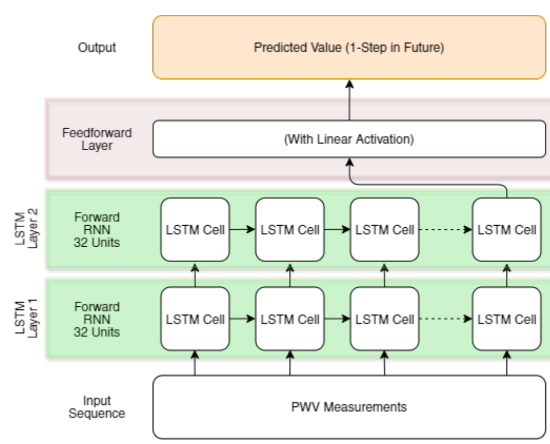

Fig. 1: LSTM network model for PWV forecasting.

The model was compiled with Adam optimizer in *Keras* using the default settings but with an especially designed learning rate (lr) schedule (cf. Eq. 3). The schedule has been determined by running various experiments with varying learning rates in an attempt to minimize the loss. Further, for robust regression, Huber loss was used as the training metric

Send correspondence to M. Jain, E-mail: mayank.jain1@ucdconnect.ie.

[1]The code is available at https://github.com/jain15mayank/PWV-Forecasts-Using-LSTM.

[5]. The model was trained for 150 epochs with a batch size of 32 on the Google Colab framework using GPU.

$$lr = \begin{cases} 10^{-4} \times 10^{epoch/20}, & \text{if } lr < 10^{-2}. \\ 10^{-2}, & \text{otherwise.} \end{cases} \quad (3)$$

We observe that adding a constant bias of -0.62 to the trained model reduces its error rate considerably. This value is noted manually after training has been completed. The reason for adding this bias is because the last layer of the network is a simple feed-forward dense layer with 1 neuron and linear activation. Hence, this bias is nothing but a minor modification in one of the network's weights itself which probably wasn't optimized properly during the training process.

III. Results & Discussions

The trained DNN model is benchmarked with two popular baselines which are used for time-series forecasting, namely, 'average method' (where the average of considered past data is predicted as the future value) and 'naive method' (where the recent most past value is copied over as the predicted future value). The LSTM-based DNN model is noted to perform better than both the other baselines for a lead-time of up to 40 minutes.

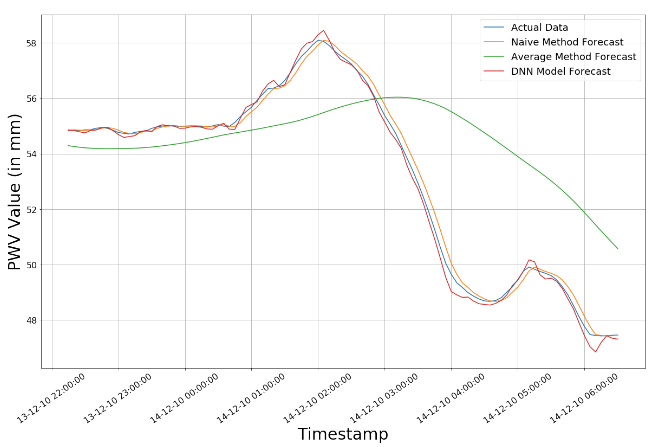

Fig. 2: Comparison of DNN (LSTM) model predictions with baselines (for 15 minutes in future).

From a qualitative perspective, the model can be noted to capture the variations in the data fairly well. This notion can be clearly seen in the figure 2 which was generated by providing real data for 4 hours before 15 minutes of the plotted value.

To quantitatively analyze the results, Root Mean Square Error (RMSE) has been calculated over the complete test set for various lead-times. The results shown in Fig. 3 that the trained DNN model performs better than both baselines up to a lead-time of 40 minutes. Moreover, with an increment in lead-time, the RMSE for DNN model also increases indicating that the error magnifies on each iteration. This is a likely possibility, as the future readings for larger lead-times were calculated using the approach of RNNLMs where the newly predicted value is assumed to be the actual value for future predictions.

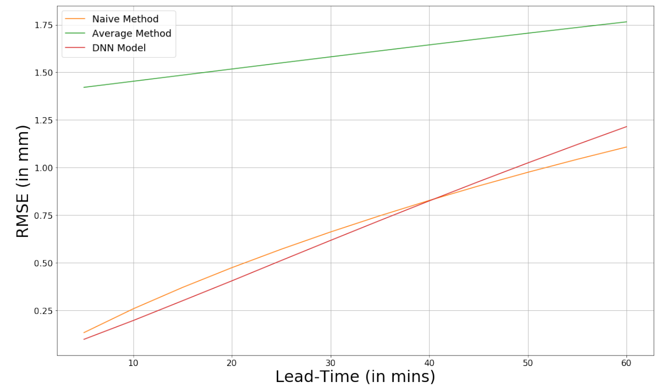

Fig. 3: RMSE values for the DNN model and baselines over a range of lead-times (5-60 minutes).

Although, the performance of the trained DNN model is not very good for larger lead-times, the network demonstrates high accuracy at short-term forecasting. Table I shows the obtained RMSE values, averaged over the entire test set (more than 1500 hours of recorded data), for varying lead times.

TABLE I: RMSE (mm) for different methods & lead-times

Lead-time	DNN Model	Naive Method	Average Method
5 min	0.0978	0.1330	1.4212
10 min	0.1966	0.2581	1.4532
15 min	0.3005	0.3704	1.4854

IV. Conclusion & Future Work

This paper presents an LSTM-based deep neural network for forecasting the future PWV values. We obtain good forecasting accuracy using our proposed framework as compared to other benchmarking methods. In the future, we intend to benchmark our LSTM-based network with other benchmarking methods [6], use longer time-period for statistical analysis, and include other sensor data [7] for better prediction.

References

[1] S. Manandhar, S. Dev, Y. H. Lee, Y. S. Meng, and S. Winkler, "A data-driven approach for accurate rainfall prediction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 9323–9331, 2019.

[2] S. Manandhar, S. Dev, Y. H. Lee, and Y. S. Meng, "On the importance of PWV in detecting precipitation," in *2018 USNC-URSI Radio Science Meeting (Joint with AP-S Symposium)*. IEEE, 2018, pp. 89–90.

[3] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.

[4] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur, "Extensions of recurrent neural network language model," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 5528–5531.

[5] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics*, pp. 492–518. Springer, 1992.

[6] S. Manandhar, S. Dev, Y. H. Lee, and S. Winkler, "Predicting GPS-based PWV measurements using exponential smoothing," in *Proc. IEEE AP-S Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*. IEEE, 2019.

[7] S. Dev, F. M. Savoy, Y. H. Lee, and S. Winkler, "Design of low-cost, compact and weather-proof whole sky imagers for High-Dynamic-Range captures," in *Proc. International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015, pp. 5359–5362.