Weather forecasting using deep learning: A comparative study

Er. Sanjay Kumar Yadav[1]*, Sankalpa KC[2]

[1]Research Supervisor: Dept. of Computer Science, St. Xavier's College, Maitighar, Kathmandu, Nepal
[2]LSRA Research Scholar: Bsc.CSIT 3rd year/5th semester, St. Xavier's College, Maitighar, Kathmandu, Nepal
Correspondence: Er. Sanjay Kumar Yadav; sanjay@sxc.edu.np*

Abstract: Accurate weather forecasting is critical for a wide range of applications, from agriculture and transportation to disaster management. In recent years, deep learning techniques such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have shown promise for improving the accuracy of various sequence-to-sequence processing tasks such as Natural Language Processing and prediction models. In this study, we conducted a comparative analysis of LSTM and GRU with encoder-decoder models using hourly MERRA-2 data to predict temperature and humidity for prediction windows of 5 hours, 10 hours, and 20 hours. Our results indicate that for prediction window of 5 and 10 hours, both the models performed similar, LSTM slightly performed better. However for the 20 hour window, LSTM performed significantly better. Our research also shows that errors in forecasted values increased with the increment of output prediction hour window.

Keywords: Weather forecasting, deep learning, RNN, LSTM, GRU, time series and MERRA-2.

1. Introduction

In many industries, such as agriculture, aviation, transportation, and disaster management, weather forecasting is essential. Although weather predicting has been done since ancient times, there have been changes in approaches, fashions, and methods.

The most substantial advancement in computational weather forecasting was the invention of Numerical Weather Prediction (NWP). NWP is based on computational models for simulating the climate by using complex differential equations relating to climate dynamics. However, it comes with its own set of hurdles.
With the advancement in internet technology, large amounts of labeled datasets have emerged. Similarly, computation power has also increased since the advent of GPUs. This, coupled with the rapidly increasing cloud computing industry, has made the idea of using machine learning very fascinating, and realistic. Weather data is a time series data, data/observations that are observed over a period of time. Weather data being a time series, demonstrates temporal dependence, which denotes that there is some relationship between past and future observations.

This relationship between past and future data/observations is the basis for prediction of time series. Machine learning is an excellent way to realize hidden patterns and relationships between data.

The weather conditions of a particular area may be different from the much wider regional weather data, as there are many uncertainties and unseen variables that affect it. In this research, we use data from a

particular location to train the machine learning models so that they can predict the weather conditions for that specific location. This research focuses on weather forecasting by implementing deep learning algorithms, using a multivariate, multi-step time series approach.

We predict the hourly temperature and humidity using past observations. Temperature (Celsius), humidity (g/kg), precipitation corrected (mm/hr), and wind speed (m/s) at 10 meters are the variables used as the input. We use two types of Recurrent Neural Networks; Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU). The models are trained for 5, 10, and 20 hour output windows with the same 24 hour input window. We use Python and the following libraries: Pandas, Numpy, Scikit-learn, Matplotlib, and Tensorflow. For the implementation of deep neural networks, we used the sequential Keras API which is integrated with Kensorflow 2.

2. Related works

A research paper (Karna et al., 2021) uses a linear regression model to predict the maximum temperature in Pokhara. Using the maximum temperature as input, the model gives its predicted maximum temperature value as the output. This is a univariate approach to weather forecasting, as only one input variable is taken into account for both input and output.

Another paper (S.Y. Ji et al., 2012) uses a rule based model for hourly rainfall prediction. It uses 13 input variables to predict and estimate the hourly rainfall. In this paper Classification And Regression Trees (CART) and C4.5 were used to make decision trees. In this paper, multiple weather input variables were taken to estimate/predict rainfall. This paper focuses on the transparency of the prediction models rather than black box approaches like neural networks.

Researchers have used the Box-Jenkins and Holt-Winters methods for forecasting meteorological time series. Murat et al.,2018 uses ARIMA and its seasonal variant (SARIMA) for daily mean temperature and daily precipitation prediction. This approach used the models separately for temperature and precipitation forecasting.

In 2009, Y. Radhika and M. Shashi published a paper that based on the daily maximum temperatures for a period of n days prior in a particular location, could forecast the maximum temperature of the next day at that site. The researchers used Support Vector Machine (SVM) and found that it consistently outperformed Multi Layer Perceptron (MLP) trained using the back-propagation approach when results were compared.

According to a review paper published in Nature (Deep Learning, Y. LeCun et al., 2015), the main advantage of deep learning is to learn complicated and nonlinear correlations between variables and outcomes.

A multivariate time-series analysis method presented in a study uses neural networks. In their studies, the price of flour was modeled and predicted using feedforward networks. The model could recognize the price curve and anticipate prices with high accuracy. These findings indicate that the neural network technique is the statistical modeling approaches' direct competitor. (Chakraborty, K. et al., 1992).

In a study, ARIMA and LSTM, two representative forecasting methods for time

series data, were evaluated for accuracy. The results showed that LSTM was superior to ARIMA when these two approaches were used for a financial dataset. In particular, the LSTM-based algorithm outperformed ARIMA by 85% on average. The research also claims that changing the number of epochs has no positive effect on the prediction accuracy of LSTM. (Siami-Namini, S. et al., 2018).

A paper by Sutskever et al. (2014) demonstrates the use of a multilayered Long Short-Term Memory (LSTM) to convert the input sequence of English language words into French words. This research uses the encoder-decoder architecture. It converts the input sequence of variable length to a fixed size vector and converts the vector to an output sequence.

A study compared regular RNN (i.e, just the tanh gate), LSTM, and GRU for the task of sequence modeling. Polyphonic music modeling and speech signal modeling were used for this research. It was found that both LSTM and GRU performed better than the traditional tanh or vanilla RNNs. However, there was no concrete decision between the performance of LSTM and GRU. (Chung J et al. 2014)

3. Methodology

3.1 Data collection

Weather data was collected from the National Aeronautics and Space Administration (NASA) Langley Research Center (LaRC) Prediction of Worldwide Energy Resource (POWER) Project, funded through the NASA Earth Science/Applied Science Program. The meteorological data is obtained from NASA's GMAO MERRA-2 assimilation model.

According to Luo B. et al. (2020), Huang et al. (2022), and Huang et al. (2021), the accuracy of weather data from the MERRA-2 dataset when compared with observed in-situ measurements was found to be relatively accurate, and a conclusion that it was fit for research purposes was drawn.

Native resolution hourly data was taken starting from 01/02/2005 until 11/15/2022. Latitude 27.6938 and Longitude 85.324 (Maitighar area) were taken as the area for weather data collection consisting of four weather variables, totaling 156,630 hourly observations:

Table 1: Data information

T2M: MERRA-2 Temperature at 2 Meters (C)
QV2M: MERRA-2 Specific Humidity at 2 Meters (g/kg)
PRECTOTCORR: MERRA-2 Precipitation Corrected (mm/hour)
WS10: MERRA-2 Wind Speed at 10 Meters (m/s) (m/s)

3.2 Data preprocessing

The dataset presented a native resolution data collected from satellites. As such no removal of outliers was done. Some preliminary preprocessing was done to the dataset; setting the date-time index and converting the object datatype of the data to numeric using the Pandas library.

The date-time index in our dataset was encoded using cyclical encoding. The date-time was encoded as sin value of hour and

day, and cos value of hour and day; resulting in 4 new variables in our dataset. This provides information about the periodic and cyclical nature of the data to the model for better understanding the relationship between them. The previous 156630 rows × 4 columns of data, after the encoding operation would transform to 156630 rows × 8 columns. The reason for choosing this encoding is that it preserves the relationship between temporal data and reduced dimensionality as opposed to other methods like one-hot encoding.

3.3 Sliding window

A sliding window protocol/methodology was used to transform the time series problem to a supervised learning problem. A sequence of training examples were created, such that a 24 hour/time-step window taken as input, had a label of 5,10 and 20 hour window.

The window "slides" across the data, so that each segment overlaps with the previous one. As such, all data points are covered. We can train machine learning models to predict the next value in the time series based on the previous values in the window.

For illustration purposes, we have created a model with 24 timestamps as the input window and 3 timestamps as the output.

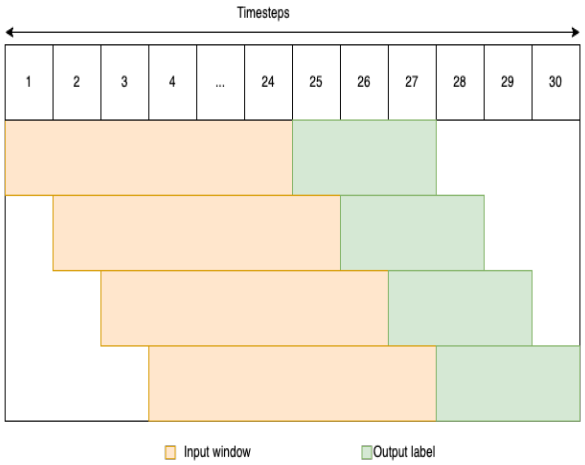


Figure 1: Sliding window model with 24 hours as input and 10 hours as output.

3.4 Data splitting

The data was split into three parts, a training set, validation set and test set. From the total of 156630 rows of data, the training set was aggregated as 0 to 135000, validation set from 135000 to 150,000 and the remaining 6630 as the test set.

3.5 Model architecture

Two types of Recurrent Neural Networks are used; Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU). Recurrent Neural Networks are . However, vanilla RNN has a fault called vanishing/exploding gradient, which makes it ineffective to learn long term dependencies.

A solution was proposed by Horchreiter and Schmidhuber; a new variant called LSTM. LSTM uses three gates: input, forget and output. LSTM could store the long term dependency from farther sequences as well as short term dependency. Again in 2014, another simpler alternative to LSTM, with only two gates, reset and update was created by Cho et al. This was the birth of GRU,

which seemed effective for sequence modeling.

These models were used in accordance with an auto-encoder. The model architecture starts with an input layer then proceeds to an LSTM/GRU layer, which encodes the data to fixed size vectors. Then the vector is duplicated according to the size of the output window. Again the duplicated vectors are passed to another layer of LSTM/GRU and finally a time distributed layer consisting of two dense layers, that specify the temperature and pressure as output variables. The models were created for 3 time windows. 5 hour output, 10 hour output and 20 hour output.

For ensuring fairness, all models were given similar conditions. Every model consists of two layers of LSTM/GRU, we had specified that the first layer should have 64 individual LSTM/GRU cells, followed by a repeat vector and the second layer of 32 LSTM/ GRU cells. Both these layers used 'relu' function as their activation function. The learning rate was specified as 0.0001 and the optimizer used was Adam optimizer. The model loss was calculated as Mean absolute Percentage Error(MAPE). The batch size was specified as 32.

To prevent overfitting of the model, we had implemented some strategies. The models are trained for a total of 200 epochs. The validation dataset was provided and the model weights were updated only if there was improvement in the validation loss. Early stopping was ensued with a patience value of 20 epochs, such that if there was no improvement of the validation loss for 20 epochs, the last best weights is saved and the training is halted.

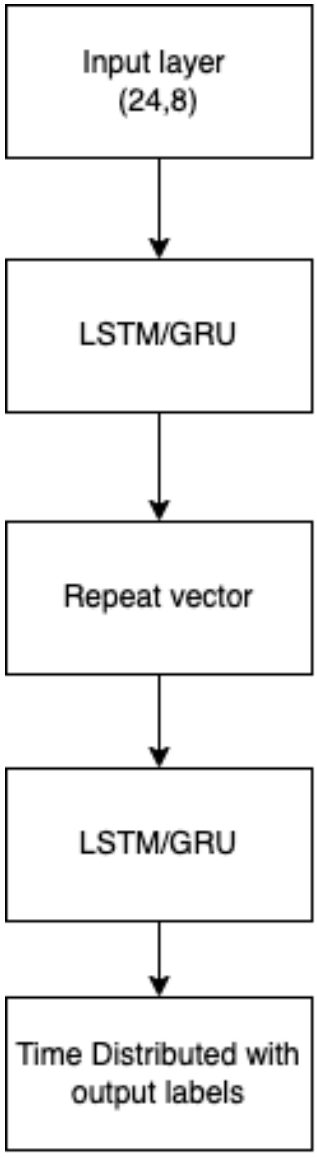


Figure 2: Model architecture

3.6 Accuracy assessment

The trained model is then fit with the test data. Its value is compared with the actual data and accuracy is determined using accuracy metrics. The accuracy is determined for both temperature and humidity values separately. Metrics used for accuracy assessment were:

Mean absolute error (MAE) =
(1/n) * ∑|actual - predicted|

Mean Squared Error (MSE): (1/n) * ∑(actual - predicted)^2

Mean Absolute Percentage Error (MAPE): (1/n) * ∑(|(actual - predicted) / actual| * 100)

The MAE indicates the absolute error of the actual vs predicted data, MSE is useful for identifying large errors/deviations. And MAPE gives an average percentage difference between actual and predicted data; ie relative error.

3.7 Model deployment

The dataset was obtained from the source, preprocessed, partitioned into windows of input dimension (24,8) and varying output of (x,y): x denotes window size (5,10,15) and y denotes number of variables (2). The dataset is then split into training, validation and test sets. The training dataset is fed into the model along with the validation set.

After the training of the models, the test dataset was then used to forecast the weather variables and its accuracy was measured by comparing the predicted values with the real values.

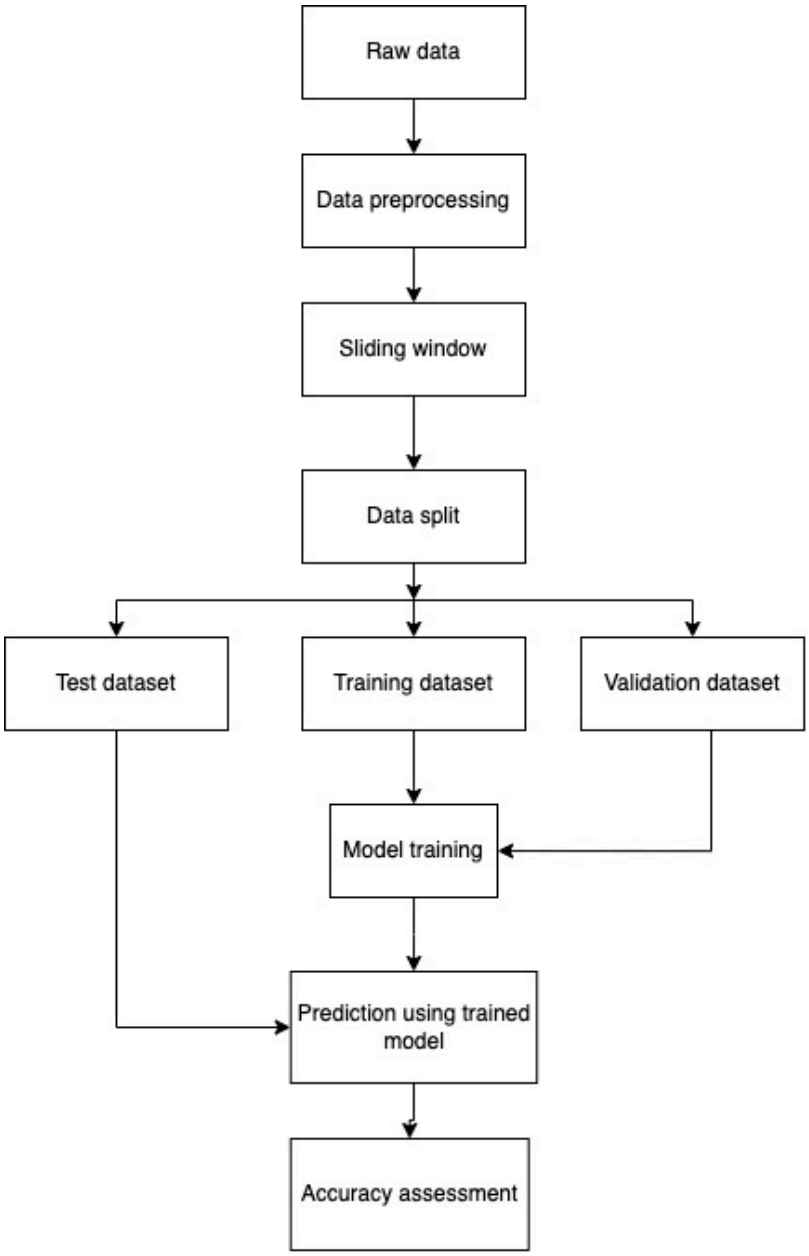


Figure 3: Methodology flowchart

4. Result

4.1 Training result

The training results were obtained as:

Table 2: Final model training and validation loss (MAPE).

Model/ window	Window 5		Window 10		Window 20	
	Train loss	Validation loss	Train loss	Validation loss	Train loss	Validation loss
LSTM	2.5251	2.8986	3.6692	4.2289	4.8546	5.9933
GRU	2.5278	2.8398	3.6829	3.9516	5.197	5.5866

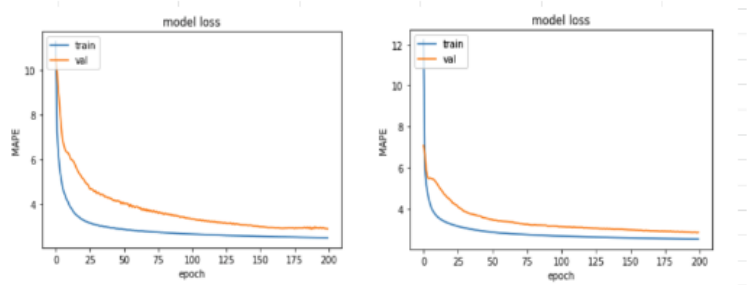
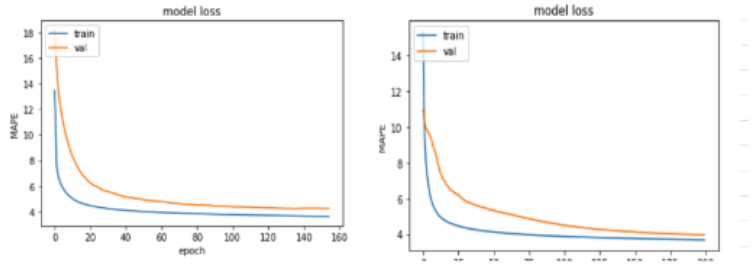
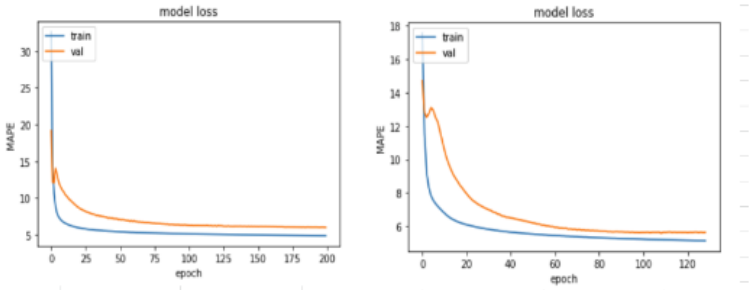
Window/Model	LSTM	GRU
Window 5		
Window 10		
Window 20		

Figure 4: Graph of training and validation loss (MAPE).

4.2 Test results

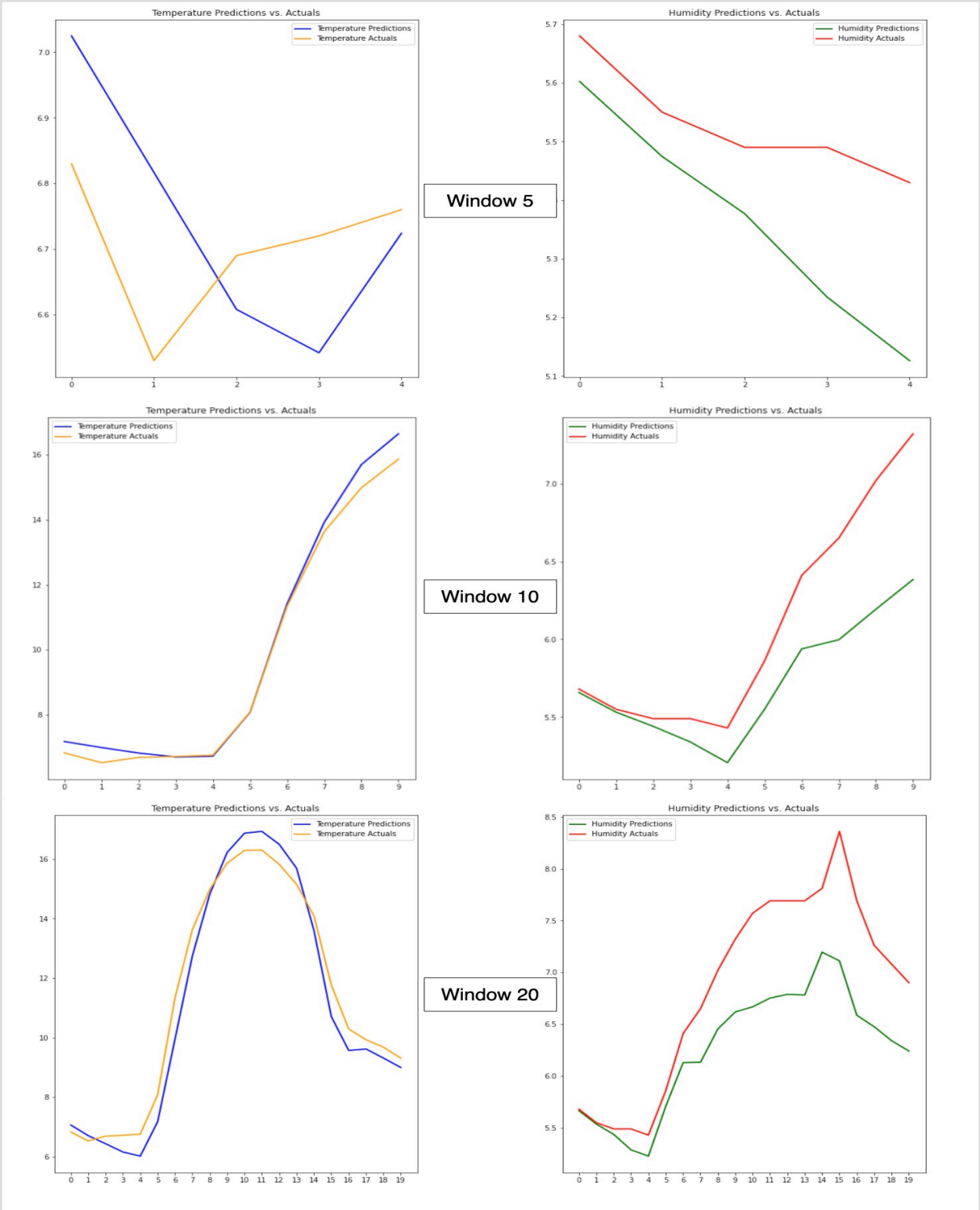


Figure 5: Comparing actual vs predicted temperature and humidity of LSTM

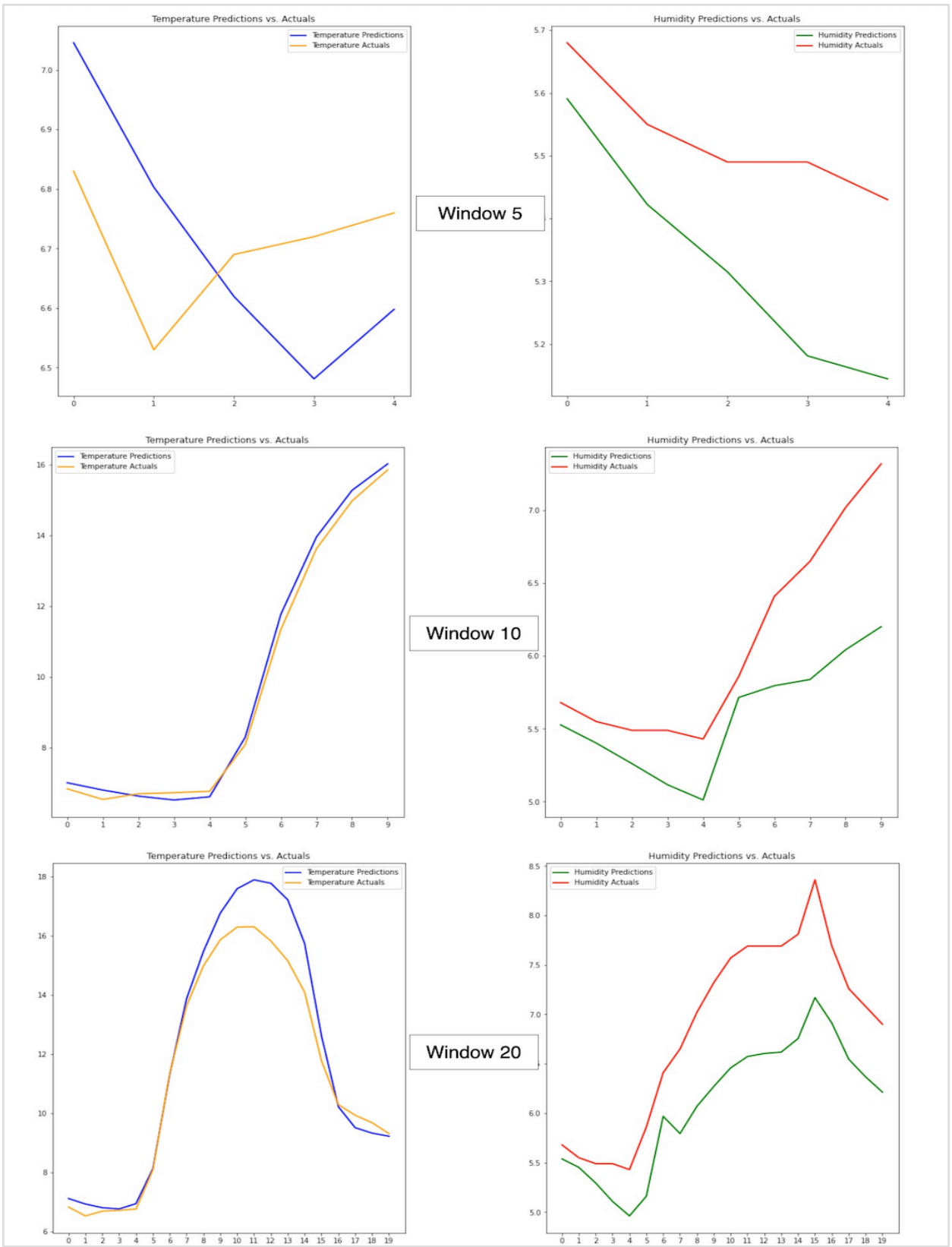


Figure 7: Comparing actual vs predicted temperature and humidity of GRU

The accuracy metrics of test data were obtained as:

Table 3: Window 5 accuracy metrics

Window 5			
	Error metric	Temperature	Humidity
	MAE	0.16	0.16
LSTM	MSE	0.03	0.04
	MAPE	2.33	3
	MAE	0.19	0.2
GRU	MSE	0.04	0.05
	MAPE	2.87	3.59

Table 4: Window 10 accuracy metrics

Window 10			
	Error metric	Temperature	Humidity
	MAE	0.29	0.37
LSTM	MSE	0.16	0.24
	MAPE	2.75	5.55
	MAE	0.23	0.5
GRU	MSE	0.06	0.37
	MAPE	2.49	7.75

Table 5: Window 20 accuracy metrics

Window 20			
	Error metric	Temperature	Humidity
	MAE	0.57	0.58
LSTM	MSE	0.42	0.47
	MAPE	5.37	7.84
	MAE	0.65	0.74
GRU	MSE	0.87	0.66
	MAPE	4.95	10.4

4. Discussion

From table 3, we found that for both LSTM and GRU, the accuracy was high. There were no large errors in both models for both temperature and humidity. LSTMs absolute and relative accuracy was slightly better for both temperature and humidity.

From table 4, we found that the values of temperature and humidity were different. When looking the temperature accuracy metrics, MSE values indicated that GRU had less large deviations as compared to LSTM. Also the absolute and relative errors for temperature was better for GRU.

For humidity, LSTM outperformed GRU in all metrics by a small difference. The MSE values indicate that LSTM had less large deviations as compared to GRU.

From table 5, we found that the values of temperature and humidity were different. When looking the temperature accuracy metrics, MSE values indicated that LSTM had significantly less large deviations as compared to GRU. Also the absolute error for temperature was better for LSTM. However, relative error was less for GRU as compared to LSTM; this indicated that there were relatively larger absolutes errors for the data forecasted by GRU, even though the rest of the forecast was relatively accurate.

For humidity, LSTM outperformed GRU in all metrics by a significant difference. Both absolute and relative errors were significantly better for LSTM. The MSE for both LSTM and GRU are high but LSTM still outshined GRU in terms of relatively less large deviated values.

The 5 hour window accuracy suggest that both models are highly accurate, however LSTM slightly outperformed GRU. GRU for the 10 hour window performed very good for temperature but was bad for humidity. In the 20 hour window, we could see that LSTM produced a consistent output with less large deviations.

5. Conclusion

By analyzing the results, we could see that the size of the output windows were directly proportional to error. Our results indicated that for the window size of hours 5 and 10, the significant difference between forecasted value from LSTM and GRU was little; however both models produced accurate enough results be useful. However, for the window size of 24, LSTM outperformed much better than GRU, indicating that LSTM could produce much more reliable forecast as compared to GRU.

6. Limitations and recommendations

The major limitations of this research were that the choice of output variables were limited to two; temperature and humidity and simplicity of model. The future prospects of this research include increasing the number of output variables and changing the model complexity by introducing varying numbers of LSTM/GRU cells and/or increasing the number of layers.

7. Acknowledgements

The authors would like to acknowledge the Father Locke and Stiller Research Award initiative and Fr. Dr. Vincent Braganza for his insights. The authors also acknowledge the Principle, Research Head, Member Secretary, Head of Department of Computer Science, Research co-ordinator of Department of Computer Science and all the research advisors. To all the helping hands, thank you for all the support and cooperation.

8. Conflict of Interest

The authors declare no conflict of interest.

9. References

Wiston, Modise & Mphale, Kgakgamatso. (2018). Weather Forecasting: From the Early Weather Wizards to Modern-day Weather Predictions. Journal of Climatology & Weather Forecasting. 06. 10.4172/2332-2594.1000229.

Martin Charlton, Alberto Caimo. (2012) Time Series Analysis. [Research Report] ESPON | Inspire Policy Making with Territorial Evidence. ffhal-03609303f.

Karna, Neha & Roy, Prem & Shakya, Subarna. (2021). Temperature Prediction using Regression Model. International Journal of Advanced Engineering.

Ji, S.-Y., Sharma, S., Yu, B., & Jeong, D. H. (2012). Designing a rule-based hourly rainfall prediction model. 2012 IEEE 13th International Conference on Information Reuse & Integration (IRI). doi:10.1109/iri.2012.6303024

Murat, Małgorzata & Malinowska, Iwona & Gos, Magdalena & Krzyszczak, Jaromir. (2018). Forecasting daily meteorological time series using ARIMA and regression models. International Agrophysics. 10.1515/intag-2017-0007.

Yalavarthi, Radhika & Shashi, M.. (2009). Atmospheric Temperature Prediction using Support Vector Machines. International Journal of Computer Theory and Engineering. 10.7763/IJCTE.2009.V1.9.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature. 10.1038/nature14539

Chakraborty, K., Mehrotra, K., Mohan, C. K., & Ranka, S. (1992). Forecasting the behavior of multivariate time series using neural networks. Neural Networks .doi:10.1016/s0893-6080(05)80092-9

Siami-Namini, S., Tavakoli, N., & Siami Namin, A. (2018). A Comparison of ARIMA and LSTM in Forecasting Time Series. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). doi:10.1109/icmla.2018.00227

Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks,"(2014) in Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS).

NASA Langley Research Center. (01/02/2005 - 11/15/2022). Prediction Of Worldwide Energy Resource (POWER) datasets. Retrieved from https://power.larc.nasa.gov/ [Accessed on [11/25/2022].

Huang, Liangke & Mo, Zhixiang & Liu, Lilong & Zhaoliang, Zeng & Chen, Jun & Si, Xiong & He, Hongchang. (2021). Evaluation of Hourly PWV Products Derived from ERA5 and MERRA-2 over the Tibetan Plateau Using Ground Based GNSS Observations by Two Enhanced Models. Earth and Space Science. 8. 10.1029/2020EA001516.

Luo, B., P. J. Minnett, M. Szczodrak, N. R. Nalli, and V. R. Morris, (2020): Accuracy Assessment of MERRA-2 and ERA-Interim Sea Surface Temperature, Air Temperature, and Humidity Profiles over the Atlantic Ocean Using AEROSE Measurements. J. Climate, 33, 6889–6909, https://doi.org/10.1175/JCLI-D-19-0955.1

Huang, L., Fang, X., Zhang, T., Wang, H., Cui, L., & Liu, L. (2022). Evaluation of surface temperature and pressure derived from MERRA-2 and ERA5 reanalysis datasets and their applications in hourly GNSS precipitable water vapor retrieval over China. Geodesy and Geodynamics, 14(2), 111–120. https://doi.org/10.1016/j.geog.2022.08.006

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), pp.1735-1780.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation." Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724-1734.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In NIPS 2014 Workshop on Deep Learning, December 2014