Using Additive and Deep Learning Algorithms for Weather Forecasting

Vivek Kakarla

Student, Independence High School, Ashburn, Virginia

DOI:10.37648/ijrst.v14i01.003

[1]*Received: 29 November 2023; Accepted: 29 January 2024; Published: 09 February 2024*

ABSTRACT

One significant element that directly impacts agricultural activities is the weather. The type of crop that should be grown depends significantly on the temperature and humidity of the area. Therefore, farmers need to be aware of upcoming weather trends to plan their farming operations. Although there are several traditional weather prediction services, they are always run by governments and rely on intricate modelling. In this scenario, time series forecasting becomes an effective method of predicting these temperature patterns because it needs historical weather data and very little computer power to provide results. In this work, we examine the steps involved in putting both additive and regression-based models into practice and evaluate their respective performances to determine which strategy is most effective for weather prediction.

INTRODUCTION

Data, including observations gathered across successive timestamps, is a time series. The primary time series analysis method is extrapolating future values from historical observations without considering other factors. Because time series may be used for various purposes, including predicting stock market indexes and real estate prices, their study and uses have gained popularity in recent years.

Weather prediction is one of the uses for time series data. Global weather conditions are changing quickly [1], and predicting these patterns is highly labour-intensive. The current method of weather prediction mainly relies on intricate physical models, and the forecasts generated by these models require extensive data gathered from various sources, including rainfall reports [2], satellite weather reports, and other sources. Government and nonprofit organisations typically make projections because the public does not have easy access to the equipment required for this. While time series prediction is not a novel concept, deep learning models and sophisticated statistical models such as ARIMA may offer a less expensive but effective substitute for forecasting future patterns in meteorological data. Furthermore, the most precise kind of forecasts—rolling forecasts of time series data—can be produced by combining these predictions with regular forecast data.

The weather is a seasonal number that repeats itself after a set amount of time, making it stationary. These seasonal trends in the weather data are used by statistical time series models such as ARIMA and Prophet and deep learning algorithms to understand intricate seasonal patterns and provide precise weather parameter predictions.

To predict the weather, we looked at various research techniques while making this article and ultimately decided on three models: Prophet, ARIMA, and LSTM. While the final method is a memory-based deep learning model, the previous two are additive. This study aids in comparing the two categories of ML model performance. We can determine whether time series approaches are substantially more affordable and dependable to create than traditional weather forecasting techniques by comparing the predicted and actual values.

MODELS

Time series forecasting using regression and deep learning models can be done in several ways. As to the survey conducted by Zhenyu Liu and colleagues [3], the most used technique for time series forecasting is ARIMA.

---

[1] How to cite the article: Kakarla V.; February 2024; Using Additive and Deep Learning Algorithms for Weather Forecasting; *International Journal of Research in Science and Technology*, Vol 14, Issue 1, 18-25, DOI: http://doi.org/10.37648/ijrst.v14i01.003

Therefore, we used ARIMA as our baseline model and compared it with our independent recommendations. ANNs, SVM, fuzzy time series for forecasting, and RNNs are a few techniques. The long-term weather parameter prediction, which LSTMs [4] have shown to be optimal, is a significant problem. Like ARIMA, the Prophet model is an additive model that offers automatic hyper-tuning parameter selection. Thus, we collectively compare the performance of implementing ARIMA, Prophet, and LSTM techniques.

A. Autoregressive Integrated Moving Average (ARIMA)

ARIMA is a set of statistical analysis models [5] that forecasts the time series results by examining the past values on the three different parameters namely Autoregression (AR), Moving Average (MA) and order of differencing (I).

1) Autoregression (AR)

The auto-regression component is the model parameter whose predictions depend on the number of past values in the time series.

2) Moving Average (MA)

The time series at any given time might be impacted by errors in various past time slots. The moving average is the model which calculates these residuals of errors in past time series and depending on that it calculates the future values.

3) Order of Differencing (I)

To apply the time series models we must first ensure the dataset is stationary. The term stationary refers to the property of the data to repeat itself in trend after periodic intervals in time.

If the dataset is not stationary, it can be made stationary by subtracting the past 't-1' values for any 't' value. The number of times this differencing is needed is the term order of differencing.

B. Prophet

It is a time series forecasting method developed by the Data Science team at Meta [6] in 2017. It is an additive model whose predictions depend collectively on namely three components: trend, seasonality and holidays.

C. Long Short-Term Memory (LSTM)

LSTM is a recurrent neural network (RNN) architecture-based deep learning model [7]. It excels at capturing long-term dependencies in the data thus making it ideal for predicting time series data which contains exhaustive historical time series data.

The different layers present in our model are:

1) Convolution Layers: These layers are responsible for extracting hidden features of our data.

2) Max Pooling and Flatten Layers: These layers are responsible for performing dimensionality reduction of our data so that the model can be trained efficiently.

3) Bi-directional LSTM Layers: These layers are responsible for knowing the forward and backward features in the sequence. It results in more accurate predictions in time series rather than normal LSTM which only focuses on past results.

4) Dense Layers: This is the end stage of LSTM and takes the output of the past LSTM layers and converts it to a higher dimension vector suitable for predicting the results.

We constructed an LSTM model consisting of 10 layers with a combination of the different layers as demonstrated in the Diagram. 1.

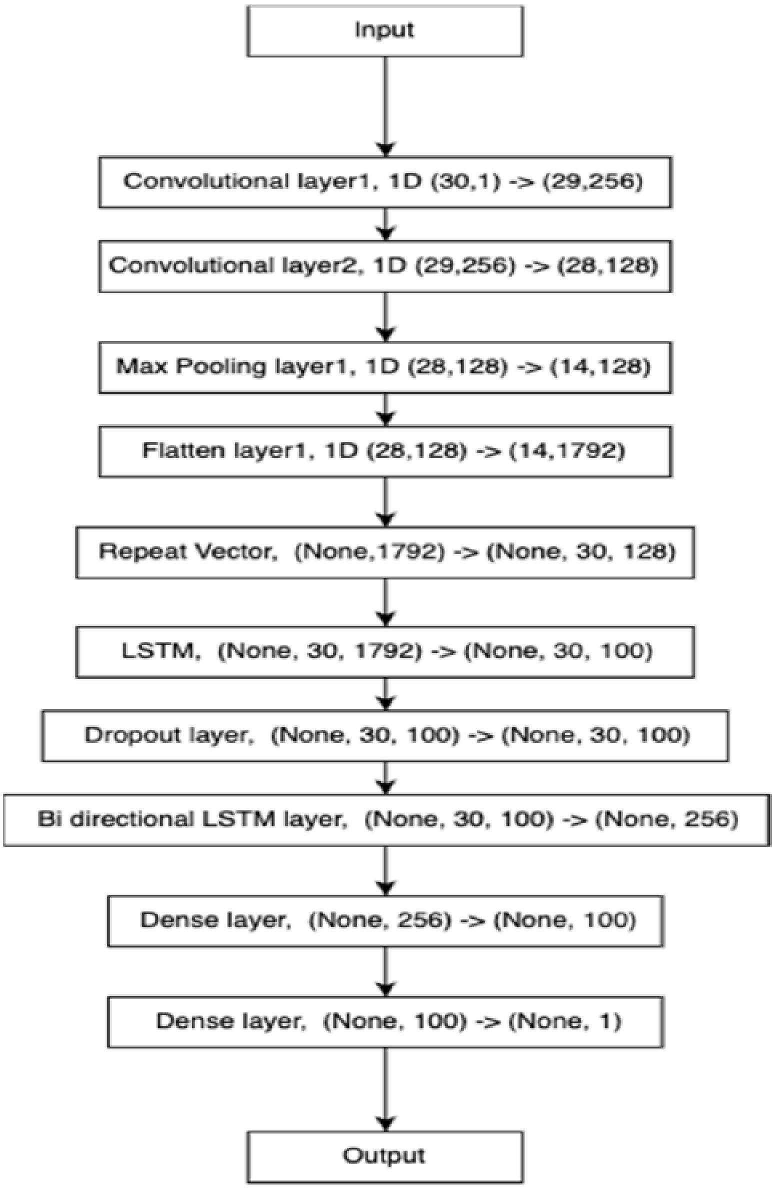

Fig 1. Layers of Bi-directional LSTM

METHODOLOGY

To train different models we study different parameters and steps necessary to train each model.

A. ARIMA

As stated in the article by ZhiQiang Li and his colleagues [8], we must first ensure that our time series data is stationary before applying ARIMA. Plotting the year data's rolling mean and standard deviation will allow you to achieve this.

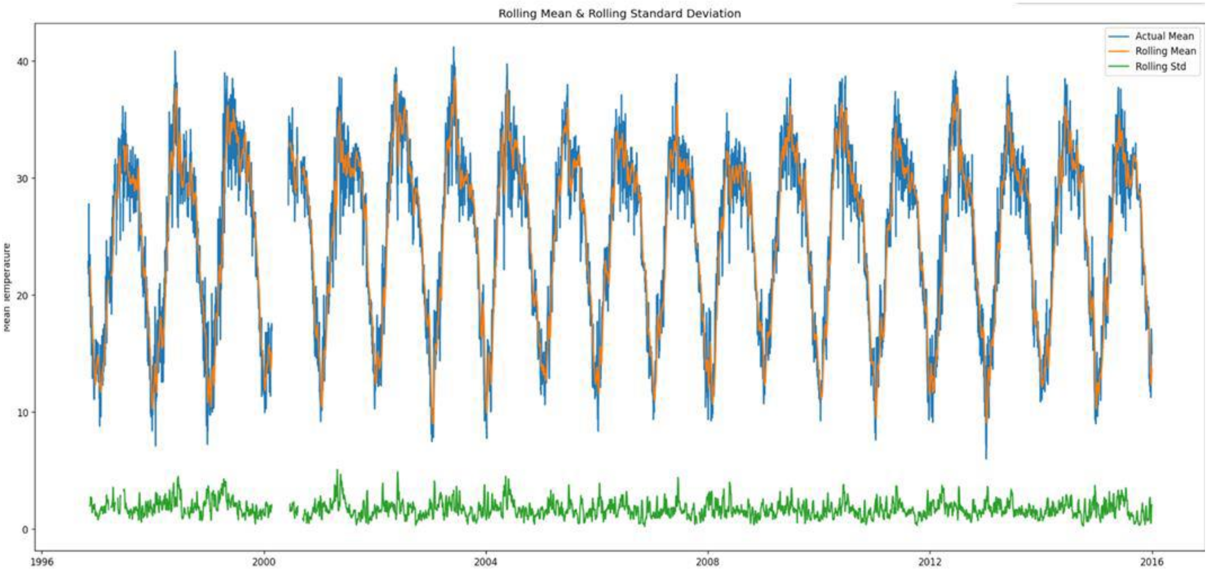

Fig. 2 Rolling Mean and Standard Deviation graph of yearly data.

The "green" line on our produced graph indicates that the standard deviation does not change over time, supporting the stationary nature of the series.

We attempted fitting the ARIMA model with AR and MA values within the 0–3 range after completing the parameter evaluation using the "prima" library to determine the lowest Akaike information criterion (AIC) score. The model's AIC score indicates how well it fits the training set. The AR term of the 3rd order, the MA term of the 1st order and the order of differencing is 0 providing us the least AIC.

TABLE I

Sr. no	ARIMA(p, d, q) parameters	AIC score
1	ARIMA(0,0,0)	72490.515
2	ARIMA(1,0,0)	30217.378
3	ARIMA(0,0,1)	44770.518
4	ARIMA(2,0,0)	30012.657
5	ARIMA(3,0,0)	29769.329
6	ARIMA(2,0,1)	29773.484
7	ARIMA(3,0,1)	29776.145
8	ARIMA(2,0,2)	29769.202
9	ARIMA(3,0,1)	29765.762

B. Prophet

Prophet has an internal framework that automatically determines the best parameters and forecasts using the given parameters.

We supply a holiday parameter with a value of 0 for our data because there are no missing data entries and a period parameter with a value of 365 days for our yearly time series data. The model itself specifies the parameters for the trend and error term.

C. LSTM

Our deep learning model is a 10-layer LSTM. The first preprocessing step in preparing the data for the model is to use the Min-Max scalar to normalise the range of features in the data. In the range of -1 to +1, this normalises the temperature and humidity time series data values. We transform the provided normalised data into a historical time series as the second step in the data preparation process. The following data in the time series is predicted using the step size and previous 30-day data points. Therefore, we create a list of time series sequences for each day entry that contains the last 30-day time series values to forecast the weather data daily.

The Dense layers are employed in our LSTM, while "Relu" serves as the activation function. We utilize the Mean Square Error function as our loss function, and the "Adam" optimizer is the one that fits the models.

Model performance is assessed using the three regression parameters: Mean Absolute Error, Root Mean Squared Error, and Mean Squared Error.

1) Mean Absolute Error (MAE): MAE depicts and calculates the difference between the variables.

2) Error of Root Mean Squared (RMSE): RMSE calculates differences between the predicted values and the actual values. The difference is the prediction error incurred during the process in any model

3) Error of Mean Squared (MSE): MSE calculates the total square of the differences between the predicted values and the actual values. The difference is the prediction error incurred during the process in any model.

RESULTS

We used the training data to fit models based on our findings. We obtained the anticipated vs. actual plots for the temperature and humidity meteorological data for each model method.

The "Blue" colour line represents the plots' "actual" values, while the "Red" colour line represents the "predicted" values.

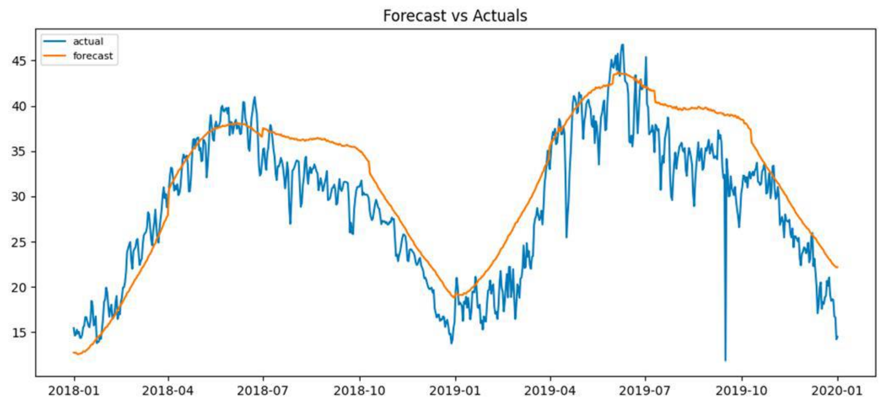

Fig. 3 Forecast vs Actual plot for temperature data using ARIMA

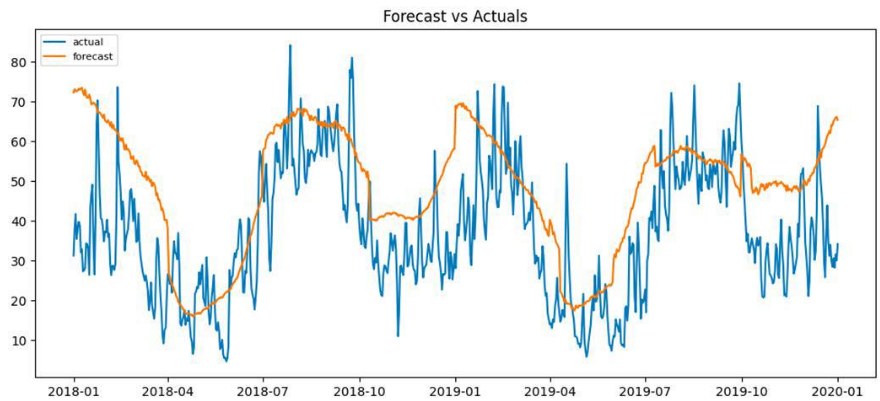

Fig. 4 Forecast vs Actual plot for humidity data using ARIMA

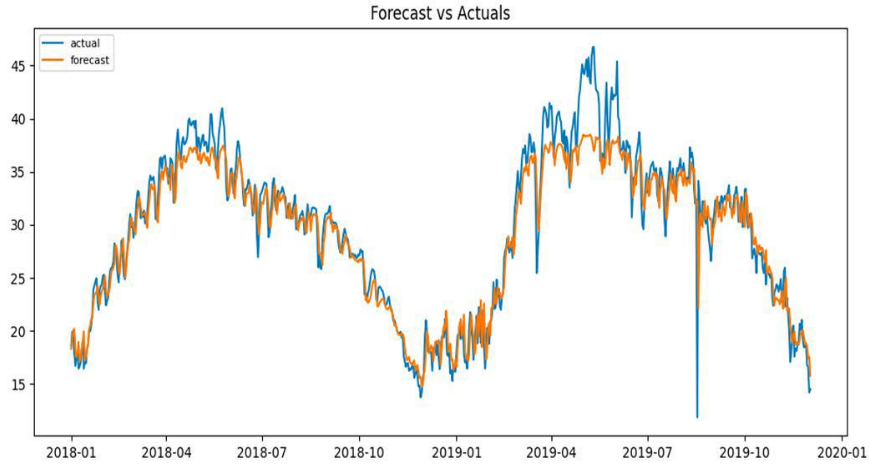

Fig. 5 Forecast vs Actual plot for temperature data using LSTM

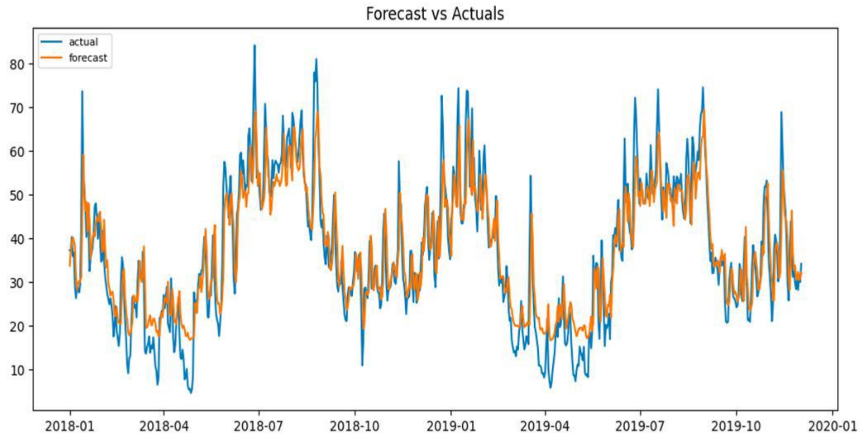

Fig. 6 Forecast vs Actual plot for humidity data using LSTM

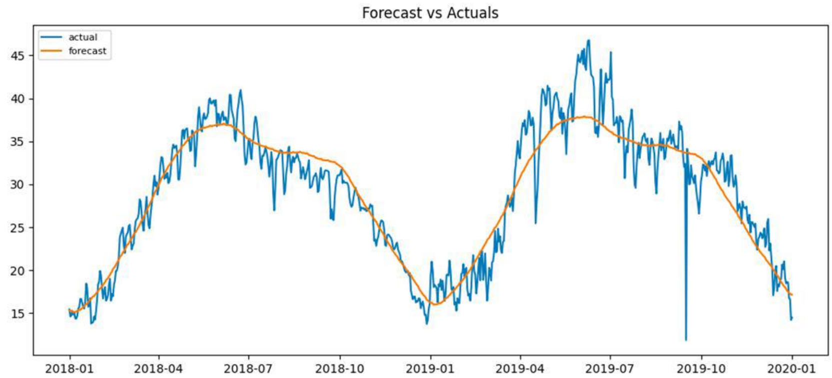

Fig. 7 Forecast vs Actual plot for temperature data using Prophet

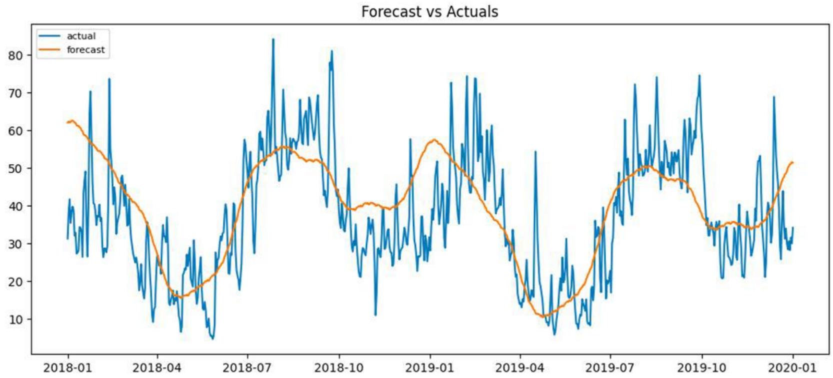

Fig. 8 Forecast vs Actual plot for humidity data using Prophet

TABLE II. Study of model performance metrics.

Model	Parameters	MAE	MSE	RMSE
ARIMA	temperature	3.346	17.283	4.157
	humidity	14.148	296.69	17.224
Prophet	temperature	2.021	7.441	2.728
	humidity	9.748	157.04	12.531
Bidirectional-LSTM	temperature	1.479	4.723	2.173
	humidity	5.126	55.578	6.676

CONCLUSION

We applied and compared regression and deep learning models for predicting temperature and humidity trends in the weather. When the plots of Figures 4 and 5 were compared to other figures, we discovered that the ARIMA model produced the least accurate forecasts. Prophet came immediately, demonstrating a substantial improvement, as seen in Figs. 8 and 9. The precise weather data readings and the Prophet model charts match very nicely. The meteorological data contains minute features that are only picked up by deep learning models like LSTM, which neither of the regression models can detect after looking at the LSTM plots in Figures 6 and 7. Deep learning models, such as LSTM, are highly developed for predicting time series trends. LSTM's granular accuracy is because, being a rolling time prediction system—generally regarded as the most accurate method of time series prediction—it continuously employs historical time sequence data to forecast future trends. The pattern is further supported by contrasting each model's performance error measures, as indicated in Table 2. With an MAE value of 1.479 degrees and 5.126 per cent relative humidity, LSTM performed the best when compared to our baseline ARIMA strategy based on these variables.

REFERENCES

1. F. V. Davenport and N. S. Diffenbaugh Using machine learning to analyze physical causes of climate change: A case study of U.S. Midwest extreme precipitation, Geophys. Res. Lett., vol. 48, no. 15, Aug. 2021 Art. no. e2021GL093787
2. Krouma, M., Yiou, P., Déandreis, C., and Thao, S.: Assessment of stochastic weather forecast of precipitation near European cities, based on analogs of circulation, Geosci. Model Dev., 15, 4941–4958, https://doi.org/10.5194/gmd-15-4941-2022, 2022.
3. Z. Liu, Z. Zhu, J. Gao and C. Xu, "Forecast Methods for Time Series Data: A Survey," in IEEE Access, vol. 9, pp. 91896-91912, 2021, doi:10.1109/ACCESS.2021.3091162

(IJRST) 2024, Vol. No. 14, Issue No. 1, Jan-Mar **e-ISSN: 2249-0604, p-ISSN: 2454-180X**

4. A. Srivastava and A. S, "Weather Prediction Using LSTM Neural Networks," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 2022, pp. 1-4, doi: 10.1109/I2CT54291.2022.9824268.
5. J. Pant, R. K. Sharma, A. Juyal, D. Singh, H. Pant and P. Pant, "A Machine-Learning Approach to Time Series Forecasting of Temperature," 2022 6th International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, 2022, pp. 1125-1129, doi: 10.1109/ICECA55336.2022.10009165.
6. Taylor SJ, Letham B. 2017. Forecasting at scale. PeerJ Preprints 5:e3190v2. doi: https://doi.org/10.7287/peerj.preprints.3190v2
7. Sepp Hochreiter, Jürgen Schmidhuber; Long Short-Term Memory. Neural Comput 1997; 9 (8): 1735–1780. doi: https://doi.org/10.1162/neco.1997.9.8.1735
8. Z. Li, H. Zou and B. Qi, "Application of ARIMA and LSTM in Relative Humidity Prediction," 2019 IEEE 19th International Conference on Communication Technology (ICCT), Xi'an, China, 2019, pp. 1544-1549, doi: 10.1109/ICCT46805.2019.8947142.
9. Kumar Prakhar, Sountharrajan S, Suganya E, Karthiga M Effective Stock Price Prediction using Time Series Forecasting IEEE 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)