Predicting Short-Term Weather Parameters with Random Forest and LSTM

Corey Zhang
University of Delaware
CISC 684-010

Abstract

This study evaluates two distinct approaches for short-term weather forecasting using the *Final Mesonet* Dataset of 12,960 observations. We use a Random Forest regressor that looks at the past six hours of temperature, humidity, pressure, and wind speed to predict the weather for the next hour. We also build a sequence-to-sequence LSTM model that uses twelve hours of the same data to forecast four hours ahead. Before training, we carefully clean the data by removing specific missing values and applying normalization to each feature. Both models are tested on the same time-based train and test split so we can compare their performance fairly. The Random Forest achieves a scaled mean squared error (MSE) of 0.1368 on one-hour forecasts, while the LSTM secures an order-of-magnitude lower scaled MSE of 0.0019 for multi-hour predictions. Training and validation curves demonstrate stable convergence without overfitting. We found that Random Forests work great for one-hour forecasts, but when you need to predict several hours ahead, a well-prepared LSTM does much better.

1 Introduction

Short-term weather forecasts are really useful, farmers use them to plan irrigation, pilots to set flight routes, and energy managers to balance renewable power. In this project, we explored two very different modeling approaches on the same data. The first is a classic Random Forest regressor, which uses the most recent six hours of temperature, humidity, pressure, and wind speed to predict exactly one hour into the future. The second is a modern sequence to sequence LSTM network, which digests a full twelve hours of those same measurements and then forecasts all four variables four hours ahead in one go. By comparing them side by side, we not only measure raw accuracy but also gain insight into how each model type handles short-term versus multi-step forecasting.

2 Data Preparation

Our starting point was the *Final Mesonet* Dataset which contains 12 960 one minute recordings of more than 75 sensor readings. To keep things focused, we zeroed in on four core features—Temp_C (air temperature), RH_2m (relative humidity), Pressure_1 (station pressure), and WndSpd_2m (wind speed). We dropped any timestamps where even one of these four readings was missing, ensuring that every six-hour and twelve-hour window we built contained complete data.

We scaled the inputs differently for each model to suit how they learn. For the Random Forest, we shifted and stretched the data so each feature (and the target) has a mean of zero and a variance of one—this prevents any variable from "hogging" the splits just because it has larger numbers. With the LSTM, we squashed every value into the 0–1 range using Min–Max scaling, which helps the network train more smoothly instead of getting pulled around by big differences in scale.

3 Model Architectures

Random Forest (6 h → 1 h)

We flattened each six-hour block of four features into a 24-dimensional input vector. Then we fed those vectors into a 100 tree *RandomForestRegressor* (wrapped in a *MultiOutputRegressor* so it could forecast all four weather variables at once). We trained on the first 80 percent of these time-ordered samples and saved the last 20 percent for testing how well it handled new data.

LSTM (12 h → 4 h)

Our LSTM network reads a 12 × 4 sequence of normalized measurements. The encoder is a 128-unit LSTM layer that compresses twelve time-steps into a single hidden state, followed by 20 percent dropout to prevent overfitting. A second LSTM decoder with 64 units then unfolds that state into four-time steps, each predicting all four variables. A final dense layer with linear activation produces the 16 outputs at once. We trained for 20 epochs using the Adam optimizer (learning rate = 0.001) and a 10 percent validation split, monitoring both MSE and MAE on the held-out validation set.

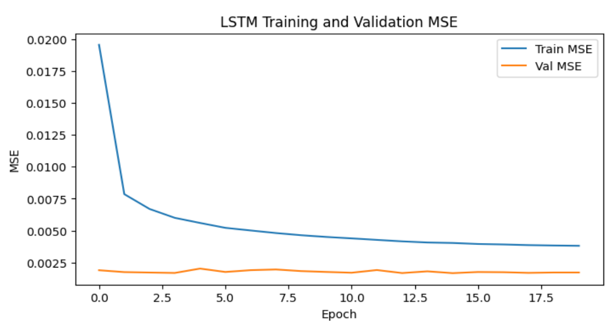
4 Results

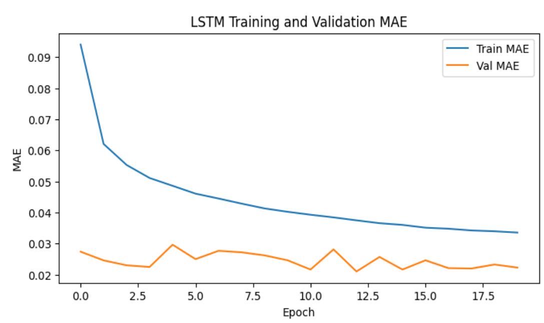
After training, we evaluated both models on **scaled** data so that their loss functions match exactly what they saw during training. Table 1 shows the test-set MSE for each approach.

Table 1. Scaled MSE Comparison

Model	Scaled MSE
Random Forest	0.136754
Seq2Seq LSTM	0.001896

Next, we examined how the LSTM's training progressed over the 20 epochs. Figure 1 plots the mean squared error on both the training and validation sets. We can see that the training MSE dropped rapidly in the first few epochs, then plateaued around 0.004, while the validation MSE settled near 0.002. Similarly, Figure 2 shows the mean absolute error for training and validation. Further confirming that adding dropout and using a modest network size provided just the right amount of regularization.





5 Discussion

Although it's a straightforward method, the Random Forest still performed well on the one-hour forecast: a scaled MSE of about 0.137 corresponds to an RMSE near 0.37 in normalized units—which works out to just a few tenths of a degree or percent once you convert back. Its limitation, though, is that it doesn't natively support multi-step forecasts—you'd need to chain one-hour predictions together to look further ahead.

Our Seq2Seq LSTM brings the MSE down to about 0.0019 for four-hour forecasts, showing that recurrent networks can learn long-term patterns well when they have enough history and the inputs are properly normalized. Also, the training and validation curves for both MSE and MAE follow each other closely, which suggests that the model size and dropout rate were chosen well. The network is powerful enough to learn the patterns while still being regularized enough to prevent overfitting.

Looking ahead, we could improve the LSTM by adding engineered features such as lagged measurements or rolling averages to give it more context. Another option is to include attention layers so the decoder can focus on the most important time steps. Even without these changes, our Seq2Seq LSTM already shows strong potential for multi-hour weather forecasting.

6 Conclusion

In this study, we compared a traditional Random Forest and a Seq2Seq LSTM for short-term weather prediction. The Random Forest is a fast and dependable baseline for one-hour forecasts, while the LSTM, when given twelve hours of normalized data, delivers accurate four-hour-ahead predictions with very low scaled error. With careful data cleaning, normalization, and thoughtful model design, we showed how both methods can work well and how deep learning can push forecast ranges further than what classical models typically handle.

Reference

xAI. (2025). *Grok 3* [Large language model]. https://grok.com/