IT1244 Project Report: Forecasting Relative Humidity

Team 30 Lim Shi Ying, Jennifer Liu, Qian Yunhan

National University of Singapore

Introduction

Relative humidity is a critical atmospheric parameter influencing weather, agriculture and daily life. Accurate short-term forecasting can support real-time decision making in areas such as disaster preparedness and farming. However, modeling relative humidity is challenging due to its nonlinear, time dependent nature. Traditional Numerical Weather Prediction (NWP) models solve complex equations to simulate atmospheric behaviour. While accurate, they are computationally intensive and slow, making them unsuitable for short-term predictions (Wiin-Nielsen, 1991). On the other hand, while Machine Learning (ML) models are efficient and can capture nonlinear dependencies well, they often operate as "black boxes," lacking interpretability (Yang, 2024).

Hence, in our project, we explored a range of interpretable ML techniques for multi-horizon relative humidity forecasting (1h, 6h, and 24h ahead), including Decision Trees (DT), Random Forests (RF), and Gradient Boosted Trees via XGBoost (XGB). These tree-based models were chosen for their balance of interpretability, efficiency, and strong performance on tabular data. We further included Long Short-Term Memory (LSTM) networks to compare against deep learning methods tailored for sequential data.

We drew inspiration from several related past works but also acknowledge some of their limitations:

• Study A – Random Forests used to model time series data often relied on raw lagged features but do not focus on model interpretability or optimal lag selection specific to the forecast horizon and typically lacked domain-specific feature engineering or pruning (Tyralis and Papacharalampous, 2017).

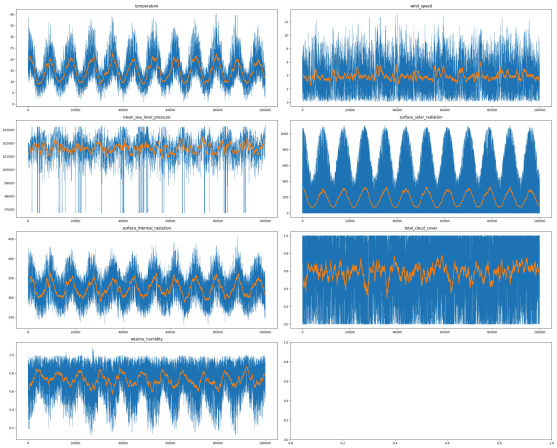
• Study B – Single-step LSTM models incorporating fourier time features to encode seasonality. However, there was no systematic hyperparameters optimization and benchmark against classical ML models. (Zhang et al., 2021)

Data Analysis

Our dataset consists of 100,057 rows and 7 weather parameters, recorded hourly from January 1, 2010, to June 1, 2021 at Monash University in Australia. Notably, there are no missing values.

Visualisations: We overlaid 2 plots - weather variables across time and its moving average over 30 days. Here, we can see surface_solar_radiation, surface_thermal_radiation, temperature and relative humidity displaying a seasonal pattern. As shown, the dataset is over 11.5 years, and hence we see 11.5 peaks. Other variables like total_cloud_cover, mean_sea_level_pressure and wind_speed do not exhibit any distinct season patterns. These season patterns could come into consideration when building our models later.

Seasonality: Many variables demonstrate seasonal patterns of daily and yearly patterns. However, the dataset is large enough such that training, validation and test sets cover all seasons to avoid bias.



Moving average over 30 days across time

Data Preprocessing

To understand the distribution of each variable, we plotted histograms (see Figure X). We observed that:

• Temperature, wind speed, mean sea level pressure, and surface thermal radiation exhibit slightly skewed normal distributions. We standardized these features to prevent potential issues like exploding gradients in neural networks.

• Created dummy variables for anomalies:

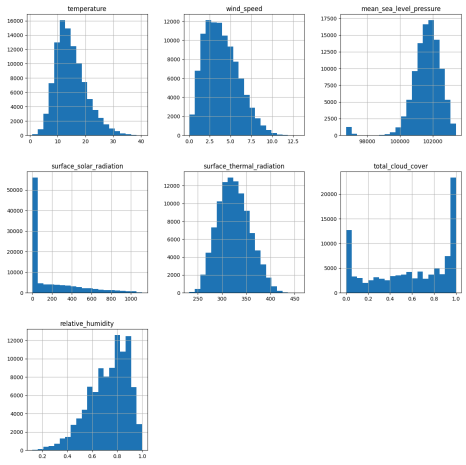
  • Mean sea level pressure below 98050

  • Surface solar radiation equal to 0 (Representing nighttime).

  • Relative humidity values exceeding 1 were rare and observed only in 2011 and 2013. While this is physically

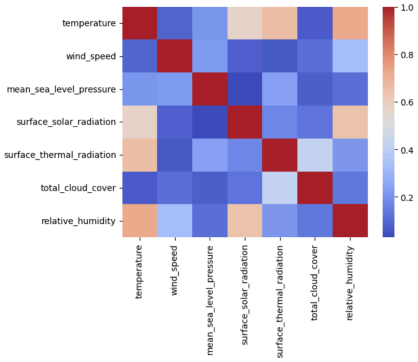
possible under supersaturation conditions, we opted to cap these values at 1 to maintain consistency.

• Engineered temporal features like hour/ week of the day and rolling averages of certain continuous variables



Histogram to examine distribution of variables

Correlation Analysis: Correlation matrix was plotted to observe the multicollinearity between features. For decision tree based models, there is no need to account for multicollinearity. However, as the collinearity is not significant, we will be keeping the features as it is. Given the large dataset and potential unknown relationship between the features, we will be keeping these 7 to allow the models to capture the non-linear relationships.

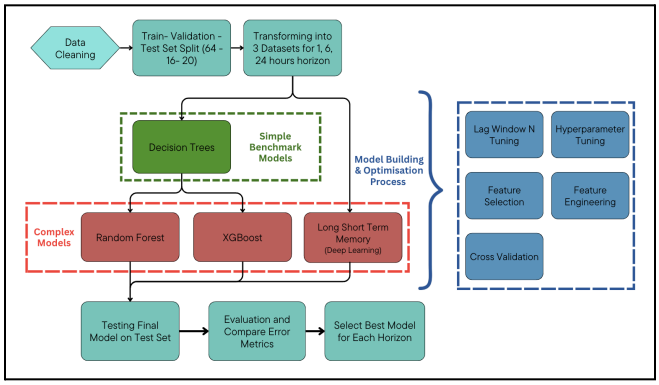


Correlation Matrix

Stationarity Test: The ADF Stationarity test was run on all variables and all features are stationary. This means that the relationships between features and target variables remain consistent throughout the dataset and there is no drift. Performing a standard 80-20 train test split sequentially will not lead to a biased evaluation.

Methodology

Our goal is to forecast the relative humidity at multiple future time spots (1h, 6h, 24h) using only past weather features. We have chosen traditional ML models (Decision Tree, Random Forest, XGBoost) and deep learning time series models (LSTM) and will compare their effectiveness.



Flow Chart

After standardizing our dataset, we split it into 64% for training, 16% for validation, and 20% for testing. We ensured that the test set must be chronologically after the training set (to avoid lookahead bias) and data points are not shuffled to preserve the chronological order to avoid data leakage. The test set is only used for final evaluation after selecting the best model for each time horizon. We then converted our features into suitable dataframes for training, and tuned the hyperparameters for each model.

Decision Trees (DTs)

To establish a baseline, we began with interpretable decision trees, which are easy to visualise. DTs recursively split the data based on feature values to minimize prediction error, using Mean Squared Error (MSE) as the split criterion by default. Each node acts like a decision rule, attempting to minimize the variance of the target variable while each leaf represents a predicted outcome. For each forecast horizon (1h, 6h, 24h), we constructed separate datasets using lagged weather features (excluding relative humidity) as input and relative humidity as the target. Each dataset used a custom lag window (lag_N) optimized per horizon based on experimentation with subsets of the data. 6 window sizes (1, 3, 6, 12, 24, 48) were evaluated. GridSearchCV with TimeSeriesSplit (5 folds) was used to search over 36 different hyperparameter combinations, optimizing tree depth, split threshold and leaf sizes. This totalled 540 experiments across the 3 forecast horizons.

Random Forest (RF)

Building on the decision tree baseline, we employed RF to improve predictive accuracy via ensemble learning. RF trains multiple decision trees on random subsets of the data and features, and aggregates their predictions. This ensemble approach enhances robustness, reduces variance, and mitigates overfitting compared to a single tree. We enriched the feature set through feature engineering with

time-based dummies (such as hour-of-day and day/night indicators) and rolling statistics. This incorporation of time-aware features beyond the basic lagged variables provides the model with richer contextual information. GridSearchCV was used to tune hyperparameters like number of trees, tree depth, samples required for splitting and features to consider. The search with TimeSeriesSplit (3 folds) to search over 16 combinations, totalling 144 experiments for 3 forecast horizons. To further improve generalization, we applied feature pruning techniques like permutation importance and cumulative contribution analysis to drop redundant or noisy predictors, streamlining model performance through dimensionality reduction. Our creation of time-aware features and pruning of features overcomes Study A's limitation of only utilising generic lagged variables without incorporating domain-specific features. Our efforts in dimensionality reduction to streamline the model mitigates the risk of overfitting, enhances computational efficiency and makes it more interpretable, which seeks to improve on Study A which evaluates predictive performance but does not emphasize model interpretability.

Gradient Boosting - XGBoost (XGB)

To capture more complex patterns in the data, we fitted similar inputs into XGB, a gradient-boosted decision tree algorithm, tuning lag windows again. Unlike RF which trains multiple de-correlated trees in parallel, XGB builds trees sequentially, with each new tree learning to correct the errors of the previous one. A change we made was using the Mean Absolute Error (MAE) as the loss function instead of MSE. Since relative humidity is normalized between 0 and 1, MAE is more interpretable and less sensitive to small outliers than MSE. XGB also adds regularization terms to penalize model complexity, helping it perform better under noisy patterns and prevent overfitting. We used Optuna, a Bayesian optimization framework, to tune hyperparameters like number of boosting rounds, tree depth, learning rate, randomness and regularization. We used 10 trials on 5 folds using TimeSeriesSplit, totalling 150 experiments.

Long Short Term Memory

LSTM is a type of deep learning model designed to handle time series data. Unlike traditional neural networks that treat each input independently, LSTMs can understand sequences over time and avoid the vanishing/exploding gradient problem by prioritising more recent data, the degree of which is controlled by a gating system. In our model, we used lagged weather data (excluding humidity) as input. This data was passed through layers of LSTM cells, each learning relationships across time steps. In hidden layers, forget gates have a sigmoid activation function whereas input and output gates have hyperbolic activation functions. Finally, a dense layer produced a single-step output: the predicted relative humidity at a specific future time (1h, 6h, or 24h). We trained the model using both MAE and MSE, and tuned key hyperparameters by manually looping through 20 random combinations of: lag window size, sampling rate, batch size, number of neurons and number of hidden layers. The optimised LSTM is denoted as LSTM1.

In addition to the base LSTM with 8 features, we introduced Fourier time features to capture the seasonality in variables like temperature, solar radiation, thermal radiation and relative humidity (LSTM2). Finally, we also implemented a multi-step prediction approach, which used LSTM to predict values of the 8 features in the next 5 or 23 time steps before using the extended data frame to perform a 1-hour ahead prediction (LSTM2).

Results

We used R^2 as our primary evaluation metric, measuring how well the predicted relative humidity values align with the actual ones, indicating how much variance in the target variable the model can explain. We also report MAE (Mean Absolute Error) and MSE (Mean Squared Error) to show the average prediction errors, but R^2 provides the clearest indicator of model fit and performance across different forecast horizons.

Tuning Lag Window N

DT	1h: lag_3	6h: lag_6	24h: lag_12
XGB	1h: lag_24	6h: lag_72	24h: lag_24

We selected lag_N by balancing performance and temporal relevance. For 1h, lag_N=3 captured recent trends with low complexity; for 6h, lag_N=6 aligned with the forecast window; and for 24h, lag_N=12 reflected diurnal patterns without introducing stale signals. This explicit lag selection tailored to the forecast horizon is unlike in Study A where they used fixed lag sets without tailoring to forecast horizons. This allowed our models to avoid outdated signals, reduce complexity, and focus only on temporally relevant features, leading to better generalisation and interpretability.

We noticed that the optimal N lags for DTs are generally smaller than that for XGB. Since Decision Trees split data based on each row independently, shorter lag windows might provide better recent signals. XGB uses sequential learning and regularization, making it more robust to longer input windows and detect deeper patterns. Interestingly, for 6h forecasts, XGB performed better with a 3 day window, as it captures momentum across multiple day-night transitions. While for 24h forecasts, XGB benefits from exactly one full-day lookback, as diurnal humidity cycles are strong.

Decision Trees, Random Forests and XGB

	DT	DT Tuned	RF Tuned	RF	XGB	XGB
1h - R^2	0.6694	0.7385	0.8033	0.8122	0.8365	0.8357
6h - R^2	0.3825	0.5635	0.6732	0.6483	0.7043	0.7028
24h - R^2	0.1194	0.4854	0.5306	0.5279	0.5204	0.5192

The untuned baseline decision tree performed poorly, especially for longer horizons, with low or negative R^2. Its lack of constraints led to deep, overfitted trees that captured noise and failed to generalise. Serving as a crucial benchmark, this baseline highlighted the impact of tuning hyperparameters and cross-validation: R^2 improved by 306.59% for the 24h forecast, alongside clear reductions in MAE and MSE. These gains stemmed from controlling tree complexity, limiting depth and enforcing stricter split conditions, to reduce overfitting.

Random forests still outperformed decision trees across all horizons. Their ensemble structure and random feature selection enabled better generalisation. Feature engineering added time-aware context, while feature pruning reduced dimensionality by over 60% without compromising performance. This pruning improved training efficiency and model robustness. After pruning, the 1h forecast improved across all metrics, while the 6h and 24h forecasts maintained strong performance with minor trade-offs.

While we expected XGB to perform much better, it only showed marginal gains over RF. Interestingly, feature pruning had a negative effect on XGB. This may be due to how low importance features are actually informative for XGB's sequential learning approach.

Long Short Term Memory

A summary of performance of each type of LSTM model:

	1h			6h			24h		
	LSTM1	LSTM2	LSTM3	LSTM1	LSTM2	LSTM3	LSTM1	LSTM2	LSTM3
MAE	0.0521		0.0543	0.0685	0.0663	0.073	0.0854	0.0909	0.0857
R^2 (MAE)	0.8279		0.8184	0.6877	0.7145	0.6362	0.5002	0.4613	0.5048
MSE	0.0048		0.008	0.0089		0.0104	0.013		0.0127
R^2 (MSE)	0.8186		0.6973	0.6609		0.605	0.5054		0.5193

For the 1-hour horizon, LSTM1 trained on MAE loss produced the best R^2 score. For the 6-hour horizon, LSTM2 trained on MAE loss produced the best R^2 score. However, for the 24-hour horizon, LSTM3 trained on MSE loss produced the best R^2 score. Short-term predictions benefited from simplicity. Intermediate predictions provided LSTM2 with structure and stimulated the interdependent relationship between various weather features, which gave it an advantage on the 6-hour horizon. However, the accumulation of error over time makes it unsuitable for even longer time horizons like 24 hours.

While LSTM3 performed poorly on the 1-hour and 6-hour horizons compared to LSTM1 and LSTM2, at the 24-hour horizon, it outperformed both, suggesting that incorporating time-based features becomes increasingly beneficial as the prediction horizon lengthens. These additional features, such as Fourier components encoding daily and yearly cycles, likely help capture long-range patterns that aren't evident from recent lagged data alone.

Final Model

Overall, we decided to use XGB as our final model choice, due to its consistency and efficient runtime. While LSTM showed competitive accuracy for the forecast horizon of 6h, it took over an hour to predict on the test set, making it impractical for deployment. Model fits evaluated with the Test set are plotted in the Appendix.

Discussions

Both Random Forest and LSTM models outperformed human capabilities by delivering fast, consistent, and data-driven forecasts. RFs offered strong accuracy with interpretability and efficiency, while LSTMs captured deeper temporal patterns.

While neither model needs to outperform humans to be useful, we see their strengths in delivering scalable, consistent, and data-driven predictions that enhance decision-making in weather-sensitive applications like agriculture, public health, and infrastructure planning. Besides, models like Random Forests offer transparency and interpretability, making the model easier to understand trust.

Our models use only public weather data, ensuring no privacy risks. They operate on objective climate variables, avoiding bias, though improving access to computational resources could help vulnerable communities. Rather than replacing experts, the models support meteorologists by automating routine forecasts. As climate change drives more erratic weather, machine learning models we explored offer essential, scalable support for predictive tasks like forecasting.

Limitations

Some other considerations we could have made include:

• Hybrid modelling: Combining statistical models with ML models to capture both linear and nonlinear interactions.

• Manual Lag Tuning Dependency in RF: our forecast performance depends on manually chosen lag_N, which was limited by computational resources we had at hand and manual analysis, which might not have captured the most informative temporal window.

• High Complexity and Training Cost: LSTM models especially require extensive tuning and computational resources, making them less practical for real-time or low-power applications.

References

Wiin-Nielsen, A. (1991). The birth of numerical weather prediction. Tellus A: Dynamic Meteorology and Oceanography, 43(4), 36–52. https://doi.org/10.3402/tellusa.v43i4.1193

Yang, R., Hu, J., Li, Z., et al. (2024). Interpretable machine learning for weather and climate prediction: A review. *Atmospheric Environment, 338*, 120797. https://doi.org/10.1016/j.atmosenv.2024.120797

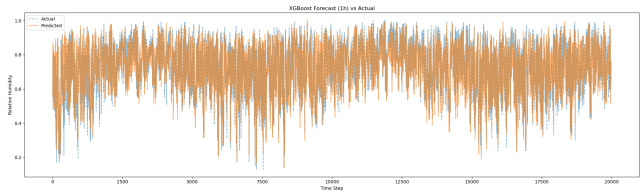
Tyralis, H., & Papacharalampous, G. (2017). Variable Selection in Time Series Forecasting Using Random Forests. Algorithms, 10(4), 114; https://doi.org/10.3390/a10040114

Zhang, W. Y., Xie, J. F., Wan, G. C., & Tong, M. S. (2021). Single-step and multi-step time series prediction for urban temperature based on LSTM model of TensorFlow. *2021 Photonics & Electromagnetics Research Symposium (PIERS)*, 1531–1535. https://doi.org/10.1109/PIERS53385.2021.9694882

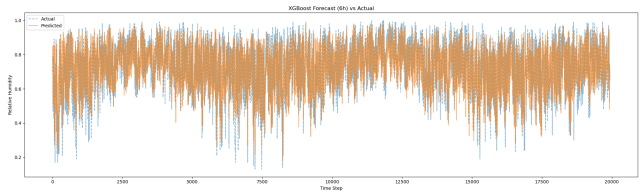
Appendix

Chosen Model: XGboost, evaluated on test set

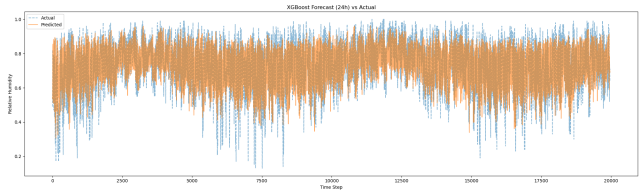
[1h] R^2 = 0.8387



[6h] R^2 = 0.7023

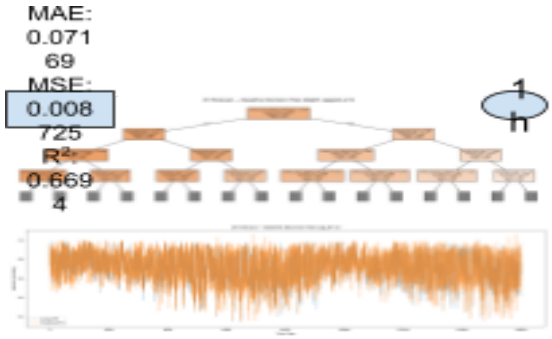


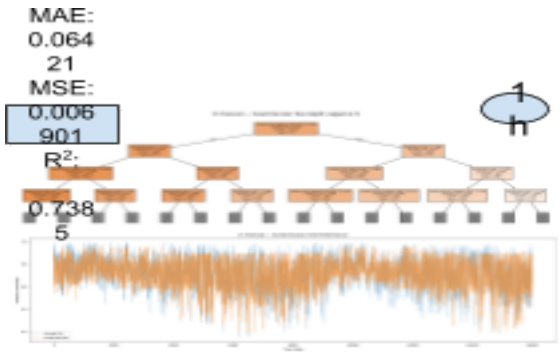
[24h] R^2 = 0.5280

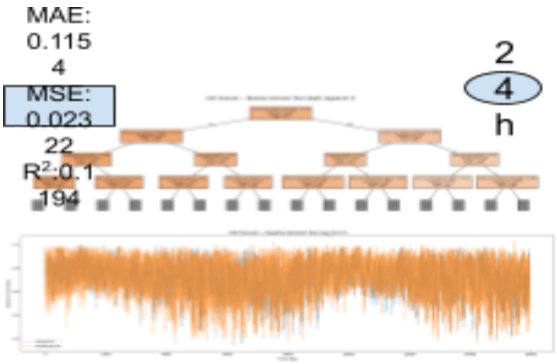


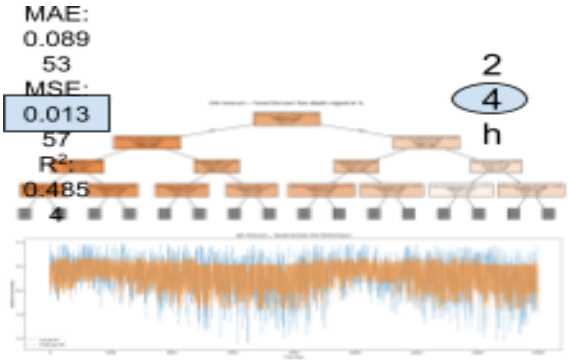
Helpful Visual Comparisons

Untuned DT 1 vs Tuned DT 2:

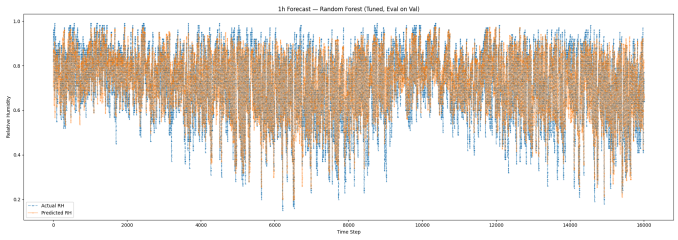




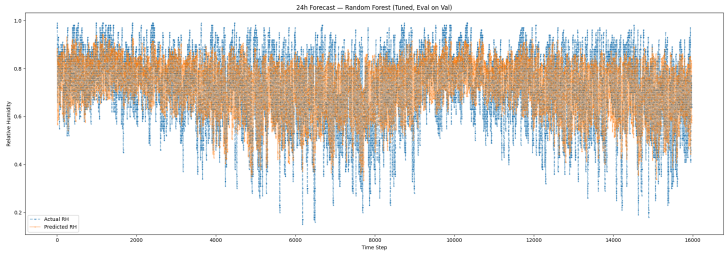




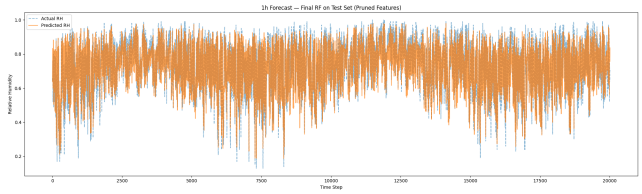
Unpruned RF 1 vs Pruned RF 2:



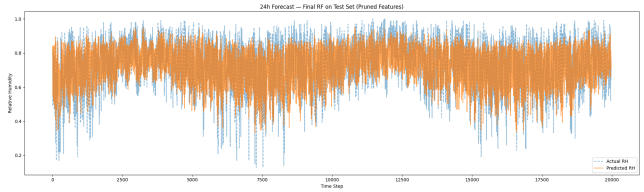
- **1h RF 1: R^2 = 0.8033**



- **24h RF 1: R^2 = 0.5306**

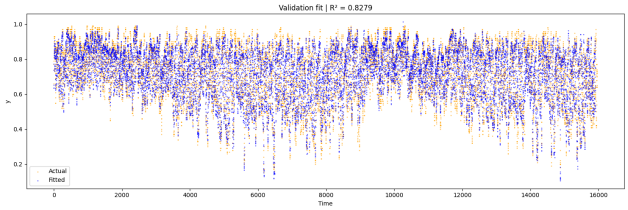


- 1h RF 2: $R^2 = 0.8122$

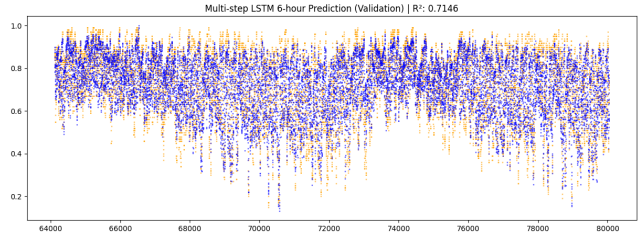


- **24h RF 2: $R^2 = 0.5280$**

1h LSTM1: $R^2 = 0.8279$



6h LSTM2: $R^2 = 0.7146$



24h LSTM3: $R^2 = 0.5197$