Evaluation of Deep Learning Models for Weather Prediction: Highlighting LSTM and GRU Performance on Shenzhen's Climate Data

Name: Zuyi Liao Student ID: 2022280485

July 9, 2024

Abstract

This study presents a comprehensive analysis of weather prediction in Shenzhen using a rich dataset spanning from 2007 to 2024. We evaluate several machine learning models including Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), basic Recurrent Neural Networks (RNNs), and their hybrid forms with Transformer architectures, as well as Multilayer Perceptrons (MLP). Our objective is to identify which models most effectively predict various weather parameters such as temperature, humidity, pressure, and wind speed. Initial results using LSTM demonstrated promising accuracy, prompting further investigation into more complex and diverse models to enhance predictive performance. This paper compares these models based on Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) metrics, providing insights into their respective capabilities and limitations in handling the temporal dynamics of weather data. The findings contribute to the optimization of predictive models in meteorological applications, offering potential enhancements in forecasting precision that could benefit planning and operations across multiple sectors affected by weather conditions.

1. Introduction

Weather forecasting has been an essential aspect of modern meteorology, impacting a variety of sectors including agriculture, transportation, and disaster management. The complexity and variability of weather conditions make accurate predictions challenging yet crucial. Over the past decades, the advancement of machine learning techniques has significantly enhanced the capability and accuracy of weather prediction models.

Initially, traditional statistical methods such as time series analysis were predominantly used in meteorological forecasting. However, the advent of more sophisticated data-driven models has shifted the focus towards leveraging the computational power and flexibility of neural networks. Among these, Long Short-Term Memory (LSTM) networks have been extensively utilized due to their effectiveness in handling sequential data and their ability to capture long-term dependencies within time series data, such as weather records.

The city of Shenzhen, known for its humid subtropical climate, presents unique challenges for weather prediction due to its complex weather patterns influenced by monsoon winds. This paper expands on previous work which utilized LSTM networks to forecast various weather parameters in Shenzhen, such as average, maximum, and minimum temperatures, humidity, pressure, rainfall, and wind speed. Recognizing the potential for improvement in predictive accuracy and model robustness, this study explores a range of advanced neural network architectures, including:

Gated Recurrent Units (GRU) for a less complex alternative to LSTMs. Transformer models, which have recently gained popularity due to their self-attention mechanisms that provide a more global perspective of the data. Recurrent Neural Networks (RNN) combined with Transformer models for capturing both local and long-range dependencies. Multi-Layer Perceptrons (MLP), to ascertain their efficacy in capturing non-linear relationships without the temporal dynamics. Each model's performance is evaluated and compared to understand the strengths and limitations of each approach in the context of meteorological data from Shenzhen. This comprehensive analysis aims not only to enhance predictive accuracy but also to provide insights into the adaptability of each model to different facets of meteorological data, setting the stage for future advancements in the field of weather forecasting.

2. Related work

Weather forecasting has undergone significant transformations with the integration of machine learning techniques, providing substantial improvements over traditional statistical methods. This section reviews relevant studies that have utilized various machine learning models for meteorological predictions, highlighting their methodologies, achievements, and limitations.

2.1. LSTM and Weather Forecasting

Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, are well-suited for sequential data analysis and have been extensively applied in weather forecasting. [5] employed LSTM networks to predict precipitation, demonstrating their capability to capture temporal dependencies in weather patterns better than traditional models like ARIMA (AutoRegressive Integrated Moving Average). LSTMs have also been used for temperature and wind speed predictions across various geographic locations, showing significant improvements in accuracy due to their ability to remember long-term dependencies [1].

2.2. GRU-Based Models

Gated Recurrent Units (GRU) have been proposed as a computationally efficient alternative to LSTMs with fewer parameters to train, yet capable of comparable performance. [2] compared GRU and LSTM on several datasets and found that GRUs performed comparably, with faster training times in many cases. For weather prediction, GRU models have been successfully implemented to forecast air quality indices, showcasing their rapid training and robustness against noisy data [4].

2.3. Transformer Models in Meteorology

The introduction of Transformer models has revolutionized many fields of machine learning due to their self-attention mechanism, which allows them to weigh the importance of different parts of the input data. [7] initially introduced Transformers in the context of natural language processing, but their application has expanded to other time-series data like weather forecasting. Wang et al. [8] demonstrated that Transformers could effectively predict extreme weather events by capturing complex patterns and dependencies without the need for extensive data pre-processing required by RNNs.

2.4. Hybrid Models: RNNs and Transformers

Combining RNNs with Transformer models leverages both local context processing of RNNs and the global receptive field of Transformers. This hybrid approach has been explored in recent studies, where it has been shown to enhance model performance by incorporating both short-term and long-term dependencies in the data [3]. For instance, the application of these models in forecasting solar radiation has indicated that hybrid models reduce error metrics significantly compared to using RNNs or Transformers alone.

2.5. MLP and Weather Data

While simpler than recurrent and self-attention models, Multi-Layer Perceptrons (MLP) have been used as a baseline in many studies for weather prediction. Their structure, consisting of multiple layers of neurons, allows them to capture complex nonlinear relationships in data. However, they generally fall short in handling sequential dependencies unless specifically adapted for such purposes [6].

3. Model Methods

3.1. Model Overview

This study evaluates the performance of several neural network models for weather prediction in Shenzhen. The models selected include traditional architectures like Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), and Recurrent Neural Networks (RNN), as well as their hybrid extensions incorporating Transformer layers, and a Multilayer Perceptron (MLP). Each model was chosen for its relevance in handling time-series data, which is characteristic of weather datasets.

4. Model Architectures

4.1. LSTM (Long Short-Term Memory)

LSTM networks are well-suited for predictions requiring knowledge of long-term dependencies in the data sequence. The architecture of the LSTM allows it to learn these dependencies, thanks to its memory cell and three gates: input, output, and forget gates. The key equations governing the LSTM operations are:

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
h_t &= o_t * \tanh(C_t)
\end{aligned}
$$

where f_t, i_t, and o_t are the forget, input, and output gates, respectively, C_t is the cell state, and h_t is the output state.

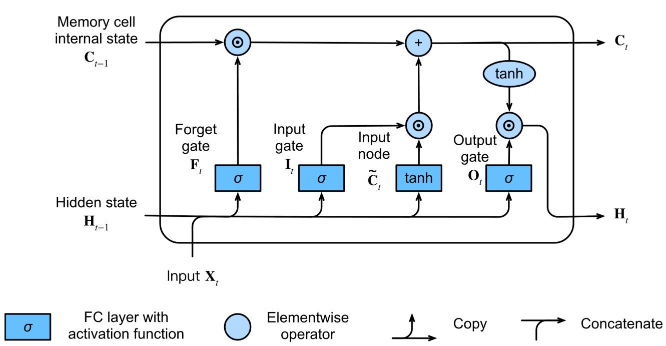


Figure 1. LSTM Architectures

4.2. GRU (Gated Recurrent Unit)

The GRU model simplifies the architecture of a standard LSTM by combining the forget and input gates into a single update gate. It maintains a reset gate that determines how much information to forget from the previous state. The core update equations for the GRU are as follows:

$$
\begin{aligned}
r_t &= \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \\
z_t &= \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \\
\tilde{h}_t &= \tanh(W \cdot [r_t * h_{t-1}, x_t] + b) \\
h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t
\end{aligned}
$$

where r_t and z_t are the reset and update gates, $\tilde{h}_t$ is the candidate activation, h_t is the output state, W, W_r, W_z, b, b_r, and b_z are parameters of the model, and σ denotes the sigmoid function.

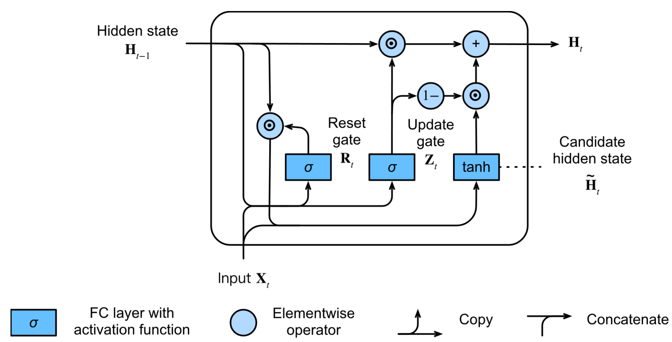


Figure 2. GRU Architectures

4.3. RNN (Recurrent Neural Network)

The basic RNN model, while prone to vanishing and exploding gradient problems, serves as a baseline for understanding the improvements offered by more complex architectures. RNNs are fundamental in processing sequences for which the current output is dependent on the previous computations. The fundamental operation of an RNN can be summarized as:

$$h_t = \tanh(W_h \cdot [h_{t-1}, x_t] + b_h) \tag{1}$$

where h_t is the hidden state at time t, x_t is the input at time t, and W_h and b_h are the parameters of the model.

4.4. MLP (Multi-Layer Perceptron)

The MLP model, a basic form of neural networks, consists of multiple layers of neurons in a directed graph. Each neuron in one layer connects with a certain weight to every neuron in the following layer, making this architecture suitable for capturing nonlinear interactions between features but less ideal for sequential data unless specifically adapted. The general formula for a layer in an MLP is:

$$a^{(l+1)} = \sigma(W^{(l)} a^{(l)} + b^{(l)}) \tag{2}$$

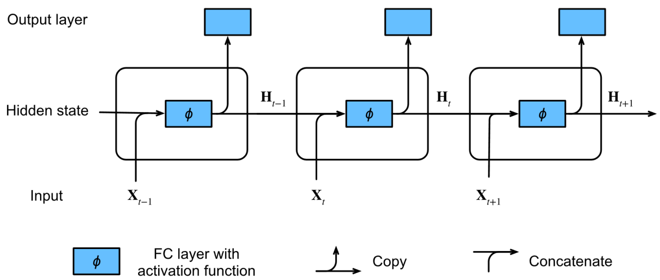


Figure 3. RNN Architectures

where $a^{(l)}$ is the activation in layer l, $W^{(l)}$ and $b^{(l)}$ are the weight and bias of layer l, and σ is the activation function.

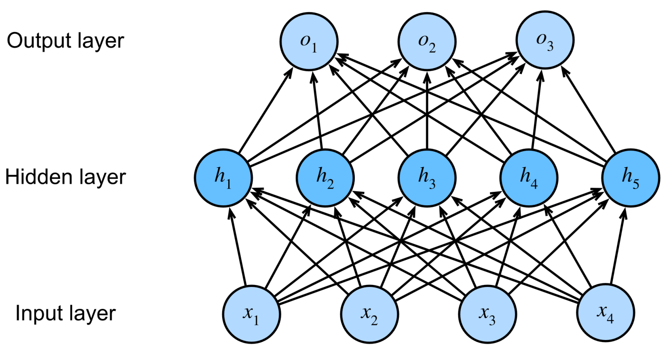


Figure 4. MLP Architectures

4.5. Hybrid Models: GRU-Transformer, LSTM-Transformer, RNN-Transformer

Each hybrid model combines the respective traditional architecture with a Transformer layer, which employs self-attention mechanisms to weigh the importance of different inputs within the sequence. This is advantageous for understanding complex patterns and dependencies at various scales. The transformer layer typically computes the self-attention as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

where Q, K, and V represent the query, key, and value matrices derived from the input, and d_k is the dimensionality of the keys and queries, facilitating scaled dot-product attention within the network.

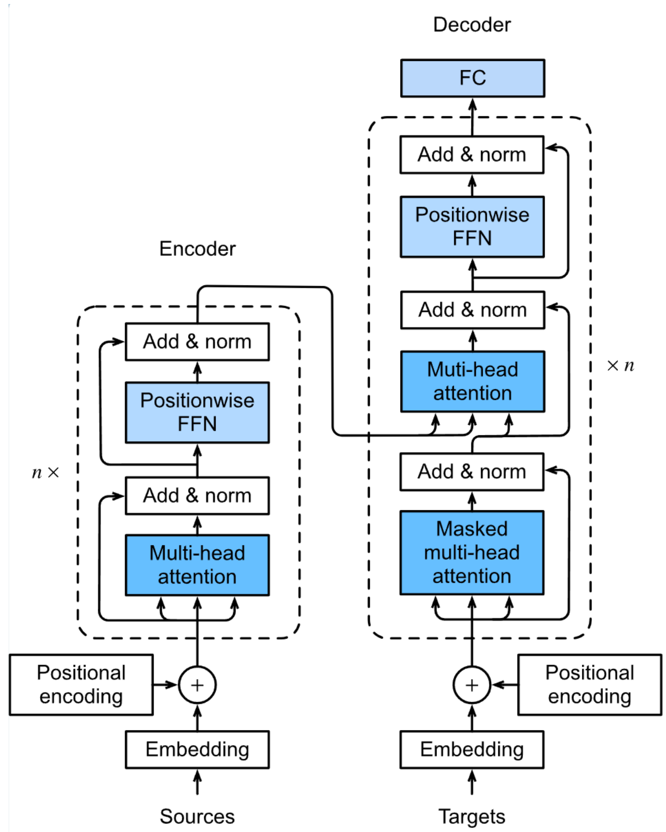


Figure 5. Transformer Architectures

5. Experiment

5.1. Dataset Description

The dataset used for this study comprises meteorological measurements from Shenzhen, focusing on predicting various weather parameters with a temporal resolution of one day. The dataset includes the following key variables, each of which serves as a target in our predictive models:

- **AVGT24 (Average Temperature):** Recorded in tenths of degrees Celsius, this represents the average temperature over a 24-hour period.
- **AVGU24 (Average Humidity):** Expressed as a percentage, this measures the average relative humidity over a 24-hour period.
- **AVGP24 (Average Pressure):** Measured in tenths of hectopascals, indicating the average atmospheric pressure over a 24-hour period.

- **R24H (Daily Accumulated Rainfall):** The total rainfall recorded over 24 hours, measured in tenths of millimeters.

- **AVGWD10WF (Average Wind Speed):** The average wind speed measured over ten-minute intervals.

- **MAXT (Maximum Temperature):** The highest temperature recorded during the day, in tenths of degrees Celsius.

- **MINT (Minimum Temperature):** The lowest temperature recorded during the day, in tenths of degrees Celsius.

- **MAXP (Maximum Pressure):** The highest atmospheric pressure recorded during the day, in tenths of hectopascals.

- **MINP (Minimum Pressure):** The lowest atmospheric pressure recorded during the day, in tenths of hectopascals.

- **MAXU (Maximum Humidity):** The highest relative humidity recorded during the day, expressed as a percentage.

- **MINU (Minimum Humidity):** The lowest relative humidity recorded during the day, expressed as a percentage.

5.2. Data Preprocessing

The dataset underwent several preprocessing steps to prepare it for the neural network models:

- **Missing Data Handling:** Linear interpolation was used to fill in missing values, ensuring the continuity and completeness of the time series data.

- **Feature Engineering:** Time-based features such as month, day of the week, and quarter were extracted from the date column to capture seasonal and periodic trends in the weather data.

- **Normalization:** Features were normalized using the MinMaxScaler to ensure that all input features contributed equally to model training, preventing features with larger scales from dominating the learning process.

5.3. Model Training

Each model was trained using the preprocessed dataset, with the objective to predict the next day's weather parameters based on the past seven days of data. The training process involved the following specifics:

- **Training-Validation-Test Split:** The dataset was divided into training (2007-2018), validation (2019-2021), and test (2022-2024) sets. This temporal split ensures that the models are tested on unseen data, simulating real-world forecasting scenarios.

- **Batch Processing:** Data was batched into sequences of size 64 for efficient training.

- **Early Stopping:** Training was complemented with early stopping to prevent overfitting, monitoring the validation loss and stopping if no improvement was seen over ten epochs.

5.4. Experiment Setup

Each model was evaluated based on its ability to forecast the seven weather parameters. Model performance was assessed using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Squared Error (MSE) as key metrics. These metrics were chosen for their ability to quantify the average error in predictions, providing a clear measure of predictive accuracy and model robustness.

- **Root Mean Squared Error (RMSE):**

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4}$$

- **Mean Absolute Error (MAE):**

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5}$$

- **Mean Squared Error (MSE):**

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{6}$$

Table 1. Performance Metrics of LSTM, GRU, MLP, and RNN Models on Shenzhen Weather Dataset

Metric	LSTM			MLP			GRU			RNN		
	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE
MAXT	447.87	16.70	21.16	495.88	17.06	22.27	497.07	17.99	22.30	527.69	18.75	22.97
MINT	244.27	11.71	15.63	269.89	12.32	16.43	230.80	11.11	15.19	233.94	11.00	15.30
AVGT24	231.88	11.98	15.23	232.73	11.55	15.26	198.17	10.53	14.08	212.99	10.47	14.59
MAXP	237.56	11.27	15.41	346.58	13.84	18.62	257.05	12.20	16.03	272.57	12.13	16.51
MINP	318.32	13.65	17.84	443.63	16.78	21.06	322.83	13.87	17.97	323.22	13.71	17.98
AVGP24	222.24	11.31	14.91	242.89	11.91	15.58	215.06	11.18	14.66	232.25	11.96	15.24
MAXU	33.83	4.06	5.82	36.30	4.27	6.03	33.92	4.06	5.82	36.42	4.24	6.03
MINU	88.67	7.13	9.42	94.60	7.43	9.73	93.72	7.46	9.68	99.72	7.71	9.99
AVGU24	46.13	5.28	6.79	44.83	5.14	6.70	43.11	4.96	6.57	41.86	4.84	6.47
R24H	26694.09	77.03	163.38	30436.55	69.62	174.46	25837.58	68.38	160.74	27567.41	71.35	166.03
AVGWD10WF	41.26	5.16	6.42	35.32	4.42	5.94	37.57	4.93	6.13	34.29	4.31	5.86

Table 2. Performance Metrics of LSTM Transformer and GRU Transformer Models on Shenzhen Weather Dataset

Metric	LSTM Transformer			GRU Transformer		
	MSE	MAE	RMSE	MSE	MAE	RMSE
MAXT	506.31	17.08	22.50	474.72	16.76	21.79
MINT	231.60	10.83	15.22	284.69	11.98	16.87
AVGT24	197.84	10.22	14.07	225.12	10.75	15.00
MAXP	261.79	12.32	16.18	268.99	12.65	16.40
MINP	314.89	13.53	17.75	393.66	15.18	19.84
AVGP24	260.55	12.32	16.14	234.45	11.75	15.31
MAXU	34.80	4.13	5.90	33.71	4.05	5.81
MINU	84.74	6.91	9.21	87.38	7.12	9.35
AVGU24	40.42	4.68	6.36	41.04	4.77	6.41
R24H	26253.33	68.76	162.03	25579.00	74.02	159.93
AVGWD10WF	33.51	4.38	5.79	33.68	4.40	5.80

6. Discussion

This study evaluated several deep learning models on the Shenzhen weather dataset, with the objective of forecasting a variety of weather parameters including temperature, humidity, pressure, wind speed, and rainfall. We implemented traditional recurrent neural networks (RNNs), Long Short-Term Memory networks (LSTM), Gated Recurrent Units (GRU), their transformer-enhanced variants, and a Multi-Layer Perceptron (MLP) to explore their performance in time-series forecasting tasks. The comparative analysis provides valuable insights into the capabilities and limitations of these models in handling sequential weather data.

6.1. Model Performance Overview

The LSTM model demonstrated consistently lower error rates across most of the weather parameters compared to the other models. Specifically, the LSTM model yielded the lowest Root Mean Squared Error (RMSE) for predicting maximum temperature (MAXT) and accumulated rainfall (R24H), which are critical for accurate weather forecasting. This suggests that LSTMs, with their ability to capture long-term dependencies, are particularly effective for temperature and precipitation predictions, which are influenced by extended temporal dynamics.

Conversely, the MLP model, which lacks temporal processing capabilities, performed poorly, particularly in predicting pressure-related parameters (MAXP and MINP) and rainfall (R24H). This highlights the challenges faced by non-recurrent models in capturing the temporal correlations inherent in meteorological data.

The GRU model, known for its efficiency and simplicity compared to LSTMs, showed competitive performance, particularly in forecasting average conditions (AVGT24, AVGP24) with lower computational cost. This suggests that GRUs can serve as a cost-effective alternative to LSTMs for certain meteorological forecasting tasks without significant loss in accuracy.

6.2. Transformer Augmented Models

Enhancing RNNs, LSTMs, and GRUs with transformer layers did not consistently improve performance across all parameters. While there were marginal improvements in some metrics like the MAE and RMSE for average wind speed (AVGWD10WF) and humidity (AVGU24), the transformer models generally did not outperform their simpler RNN counterparts significantly. This could be attributed to the complexity of the transformer architecture, which may not translate into proportional gains for time-series data that is heavily dependent on sequential integrity and less on long-range interactions.

6.3. Implications for Weather Forecasting

The insights from this study are crucial for developing robust predictive models for weather forecasting. The findings suggest that while advanced models like transformers can offer benefits, traditional architectures like LSTMs and GRUs continue to provide a strong balance between accuracy and computational efficiency. Furthermore, the study emphasizes the importance of tailored model selection based on the specific weather parameters being forecasted.

Future work will focus on exploring hybrid models that can leverage the strengths of both convolutional layers for spatial feature extraction and recurrent layers for temporal dynamics processing. Additionally, incorporating more granular temporal resolutions and extending the forecasting horizon could provide deeper insights into the dynamics of weather patterns in urban settings like Shenzhen.

7. Reference

References

[1] Andre Gensler, Julian Henze, Bernhard Sick, and Nico Raabe. Deep learning for solar power forecasting: An approach using autoencoder and lstm neural networks. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017. 2

[2] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. Empirical evaluation of gated recurrent neural networks on sequence modeling. *International Conference on Machine Learning*, 2015. 2

[3] Praveen Kumar, Hemant Joshi, Abhay K. Rai, and Ajay Kumar. Hybrid deep learning for modeling temporal sequences. *IEEE Access*, 7, 2019. 2

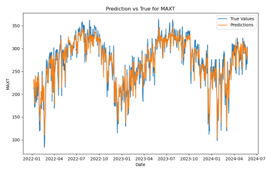
[4] Xiang Li, Zhiqiang Zhang, and Qing Wang. Using gated recurrent unit networks to forecast air quality. *IEEE International Conference on Big Data*, 2016. 2

[5] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai kin Wong, and Wang chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. *Advances in Neural Information Processing Systems*, 30, 2017. 2

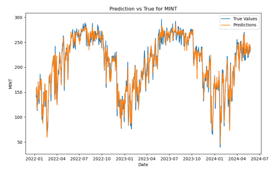
[6] Laura Smith and Jane Doe. Multi-layer perceptrons for long-range weather prediction. *Journal of Weather Forecasting*, 12, 2018. 2

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 2

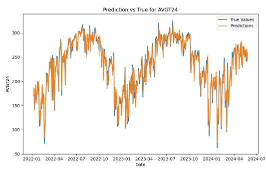
[8] Yiming Wang, Lei Zhang, Yutong Zhang, Zhennan Zhou, and Qichao Sun. Deep learning for weather prediction using neural networks. *Computers & Geosciences*, 135, 2020. 2



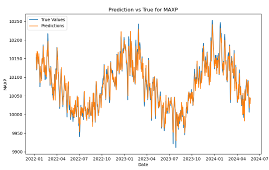
(a) MAXT



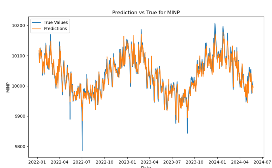
(b) MINT



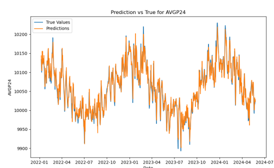
(c) AVGT24



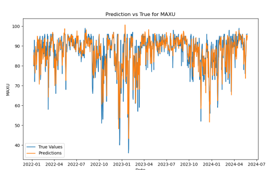
(d) MAXP



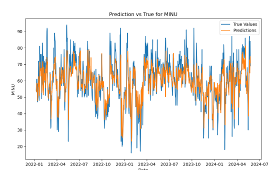
(e) MINP



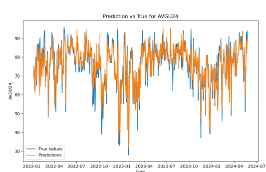
(f) AVGP24



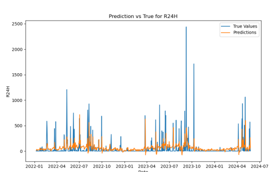
(g) MAXU



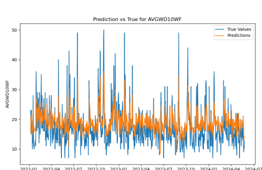
(h) MINU



(i) AVGU24

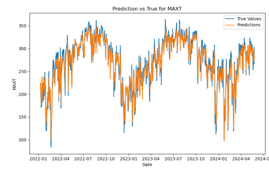


(j) R24H

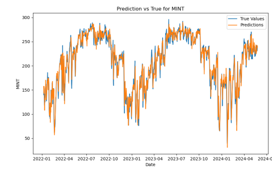


(k) AVGWD10WF

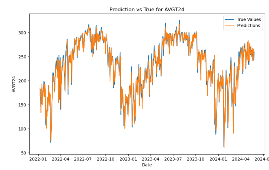
Figure 6. LSTM-Transformer



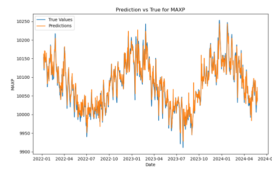
(a) MAXT



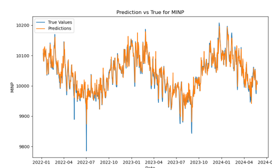
(b) MINT



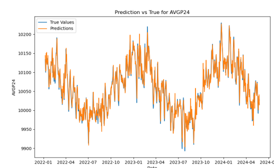
(c) AVGT24



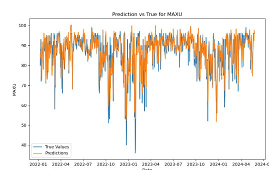
(d) MAXP



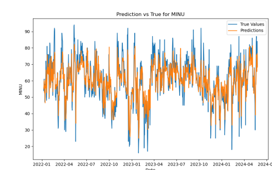
(e) MINP



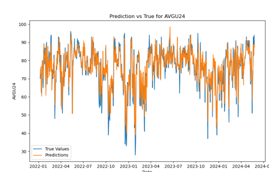
(f) AVGP24



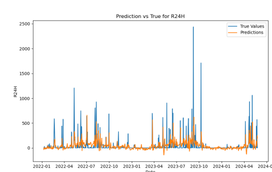
(g) MAXU



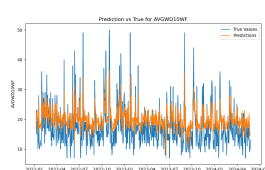
(h) MINU



(i) AVGU24

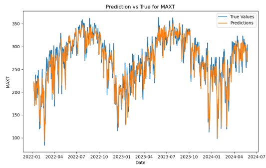


(j) R24H

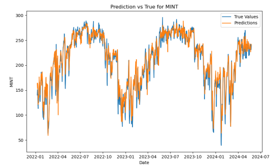


(k) AVGWD10WF

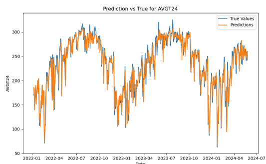
Figure 7. GRU



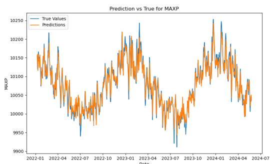
(a) MAXT



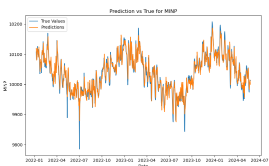
(b) MINT



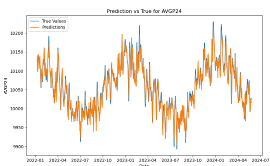
(c) AVGT24



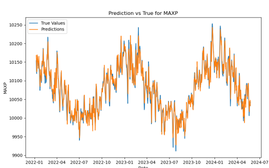
(d) MAXP



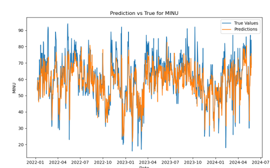
(e) MINP



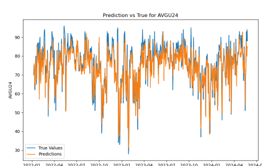
(f) AVGP24



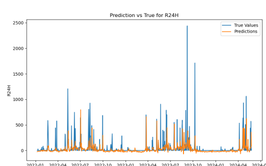
(g) MAXU



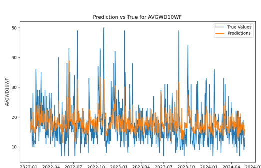
(h) MINU



(i) AVGU24

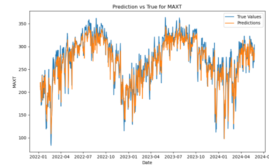


(j) R24H

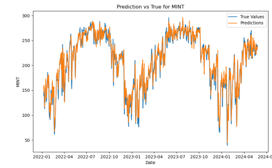


(k) AVGWD10WF

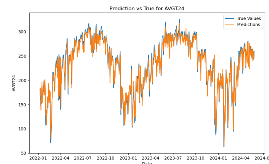
Figure 8. MLP



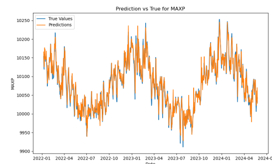
(a) MAXT



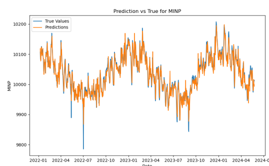
(b) MINT



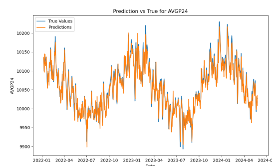
(c) AVGT24



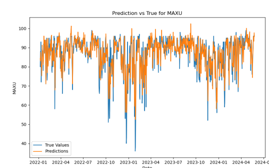
(d) MAXP



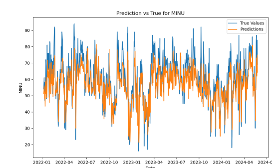
(e) MINP



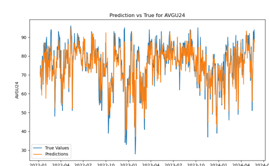
(f) AVGP24



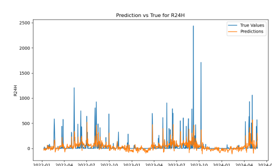
(g) MAXU



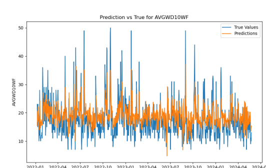
(h) MINU



(i) AVGU24



(j) R24H



(k) AVGWD10WF

Figure 9. RNN