Leveraging LSTM Neural Networks for Accurate Temperature Forecasting: A Case Study

---

Abstract

Temperature forecasting is critical in fields such as climate research, agriculture, and urban planning. Traditional statistical models often need help capturing temperature data's nonlinear and complex patterns. To address these challenges, this paper explores the application of Long Short-Term Memory (LSTM) networks, a specialized form of recurrent neural networks. Using real-world meteorological datasets, the study benchmarks LSTM against traditional models such as Random Forest, demonstrating significant improvements in forecasting accuracy. Results indicate that LSTM can effectively capture temporal dependencies and trends, making it a robust tool for temperature prediction. Practical implications of the findings are discussed, highlighting the potential of deep learning in advancing forecasting methodologies.

Keywords

Temperature forecasting, LSTM, time-series analysis, deep learning, climate modeling

---

1. Introduction

Temperature prediction is a cornerstone of modern climate science, with applications ranging from agriculture to urban development and disaster management. Accurate forecasting enables better planning, resource allocation, and response to weather-related events. However, temperature data is inherently complex, characterized by nonlinear patterns and dependencies across time scales, making it challenging for traditional statistical methods like ARIMA to provide reliable predictions.

Recent advancements in deep learning offer promising alternatives, with Long Short-Term Memory (LSTM) networks emerging as a robust tool for time-series forecasting. LSTMs are well-suited to capture long-term dependencies and nonlinear relationships, making them ideal for analyzing meteorological data. This study aims to explore the efficacy of LSTM networks in forecasting temperature, benchmarking their performance against traditional models, and identifying key areas of improvement.

---

2. Related Work

Temperature forecasting has traditionally relied on statistical approaches such as Autoregressive Integrated Moving Average (ARIMA), Seasonal Decomposition of Time Series (STL), and exponential smoothing. These methods excel in linear and stationary time-series data but struggle to model the nonlinear and non-stationary characteristics commonly observed in temperature datasets. Recent studies have highlighted these limitations, especially in their inability to capture long-term temporal dependencies and sudden changes in climatic conditions.

Machine learning approaches, including Support Vector Machines (SVM), Random Forests, and Gradient Boosted Trees, have been applied to address these challenges. These methods offer improved performance by leveraging nonlinear relationships and feature interactions. However, they require extensive feature engineering and often fall short in capturing sequential dependencies over time.

Deep learning, particularly Recurrent Neural Networks (RNN) and their advanced variants like LSTMs and Gated Recurrent Units (GRUs), has revolutionized time-series forecasting. LSTMs, in particular, have demonstrated superior performance in capturing long-term dependencies and handling missing or irregular data points. Several studies have successfully applied LSTM for forecasting stock prices, energy consumption, and meteorological parameters, underscoring its flexibility and robustness.

This paper builds on this body of work by focusing on the application of LSTMs to temperature forecasting, benchmarking their performance against traditional statistical models and machine learning methods. The review highlights the need for further exploration of LSTM architectures and hyperparameter tuning to maximize forecasting accuracy in this domain.

---

3. Methodology

3.1. Data Collection

We utilized the Jena Climate dataset from Kaggle for our analysis. This dataset comprises meteorological data collected in Germany over an eight-year period, with measurements recorded at 10-minute intervals. It includes 15 variables covering aspects such as temperature, humidity, air density, wind speed, and wind direction.

Taking a more in depth look of our dataset, we visualized the distribution of values for each variable. As shown in Image 1, most variables exhibit approximately normal distributions, though some display noticeable skews. However, certain graphs stand out more than others. For instance, the wind speed variable consistently shows values of zero, indicating that this variable may have limited usefulness for predicting weather forecasts.

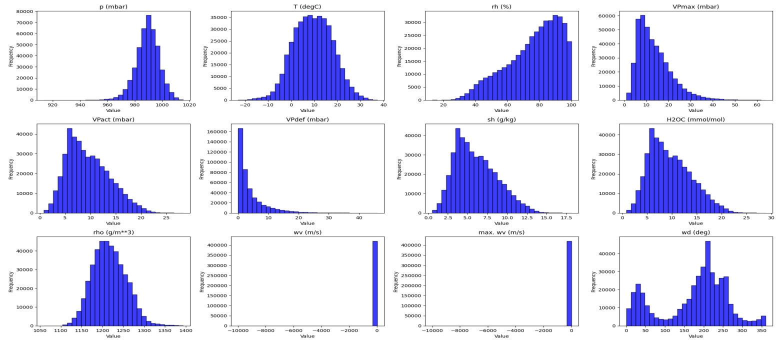

Image 1: EDA: Exploring the distributions of independent variables.

Another important aspect to examine is the relationship between variables. Image 2 presents a heatmap of correlations, highlighting how one variable might influence another. Distinct patterns emerge from this visualization. For instance, variables related to wind appear to have little to no correlation with other variables, reinforcing our earlier observation from Image 1 that these variables are unnecessary and could potentially overload the model. Additionally, we notice that rho, representing air density, has an inverse relationship with nearly every other variable. This aligns with expectations, as increased humidity means more water vapor in the atmosphere, displacing air. Lastly, variables related to humidity generally show positive correlations with most other variables, further emphasizing their interconnectedness.

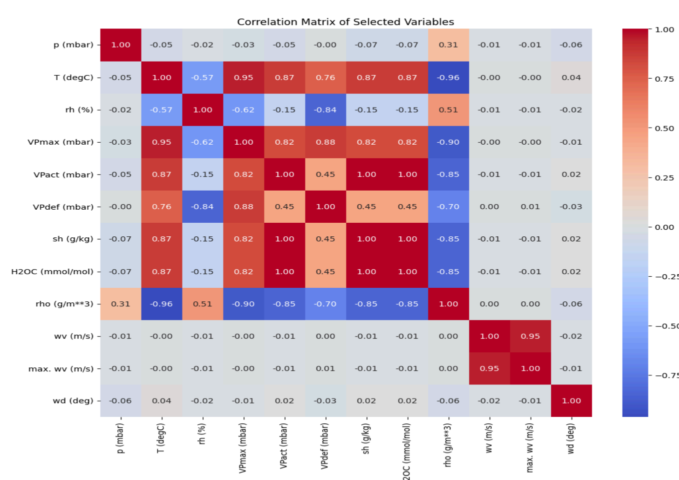

Image 2: Correlation Matrix of our variables.

3.2. Preprocessing

For preprocessing, we sorted the data in chronological order to account for the time-dependent nature of weather forecasting. The time-date variable was converted into a proper datetime format to enable efficient sorting. Given the large size of the dataset, we aimed to minimize unnecessary noise that might not contribute to the model's performance. Since temperature typically changes minimally over short intervals, we reduced the dataset by sampling data only at the start of each hour, effectively reducing its size to one-sixth of the original. Additionally, we excluded variables that were either highly correlated with the target variable or irrelevant to the prediction task. These included 'Tpot (K)', 'Tdew (degC)', 'Date Time', and 'Wind speed/direction'. After cleaning the dataset, we scaled the inputs for the training, validation, and testing sets to ensure consistency and equal contribution from all features, avoiding any single variable disproportionately influencing the model.

3.3. LSTM Model Architecture

For our approach, we primarily utilized LSTMs with ReLU activations, as these architectures are particularly effective at handling sequential data and controlling how information from past forecasts contributes to future predictions.

3.4. Training Process

To optimize performance of our models, we tested with various hyperparameters, including learning rates, batch sizes, number of hidden layers, and nodes, to optimize efficiency and accuracy. During training, we implemented early stopping, halting the process if no improvement was observed in the validation loss for three consecutive epochs. This is to prevent overfitting while reducing computational overhead. The mean squared error was used as a loss function, adam for optimizer, and various metrics for evaluation such as RMSE and MAE.

4. Experiments and Results

4.1. Experimental Setup

Model 1: 4 LSTM layers with 512, 256, 128, 64 nodes respectively and ReLu activations; 2 Dense layers with 32, 1 nodes. Predicts next hour of temperature given 24 hours.

Model 2: 3 LSTM layers with 128, 64, 32 nodes respectively and ReLu activations; 1 Dense layers with 1 node. Predicts next hour of temperature given 24 hours.

Model 3: 3 LSTM layers with 256, 128, 64 nodes respectively and ReLu activations; 2 Dense layers with 32, 1 nodes. Predicts next hour of temperature given 24 hours.

Model 4: Random Forest predicting next 6 hours given previous 24 hours

Model 5: LSTM predicting variations of next 6,12, and 24 hours given previous variations of 24, 36, and 48 hours. See appendix for details on model.

4.2. Results

Quantitative results:

- Model 1 LSTM with Relu Activations

```
{'Train R^2': 0.9870440956678982,
 'Validation R^2': 0.9845206592708403,
 'Test R^2': 0.9850242513570441}
```

- Model 2 LSTM with Relu Activations

```
{'Train R^2': 0.9912610675409552,
 'Validation R^2': 0.988182616464493,
 'Test R^2': 0.989701535259075}
```

- Model 3 LSTM with Relu Activations

```
{'Train R^2': 0.9943762065119899,
 'Validation R^2': 0.9901531310741147,
 'Test R^2': 0.9926708720528092}
```

- Model 4 Random Forest Forecasting model

```
6-Hour Horizon RMSE: 2.634737774594934
6-Hour Horizon R-squared: 0.8940058126143643
```

Model 5, 6 hour forecast given previous 24.

```
Predicted next 6 hours temperatures: [[6.84 6.63 6.38 6.13 5.92 5.75]]
Actual next 6 hours temperatures : [6.61 6.55 6.47 6.39 6.43 6.15]
Difference between predicted and actual: [[ 0.23000015  0.08000011 -0.08999989 -0.25999989 -0.50999992 -0.4
Average difference between predicted and actual: 0.2616666603088378
329/329 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step
Test RMSE: 1.4726618493329733
Test R^2: 0.9641538374681126
```

Model 5, 12 hour forecast given previous 24.

```
Predicted next 12 hours temperatures: [[7.71 7.78 7.8  7.8  7.75 7.61 7.43 7.22 7.02 6.82 6.69 6.58]]
Actual next 12 hours temperatures : [7.49 7.48 7.4  7.33 6.79 6.61 6.55 6.47 6.39 6.43 6.15 6.26]
Difference between predicted and actual: [[0.22000004 0.30000021 0.40000019 0.47000019 0.96       1.00000013
  0.87999983 0.74999979 0.62999998 0.39000017 0.54000006 0.31999992]]
Average difference between predicted and actual: 0.5716667095820108
329/329 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step
Test RMSE: 1.9572655439234006
Test R^2: 0.9366623089770423
```

Model 5, 12 hour forecast given previous 36.

```
Predicted next 12 hours temperatures: [[6.57 6.42 6.26 6.1  5.91 5.69 5.49 5.27 5.1  5.   5.   5.1 ]]
Actual next 12 hours temperatures : [7.4  7.33 6.79 6.61 6.55 6.47 6.39 6.43 6.15 6.26 6.08 5.86]
Difference between predicted and actual: [[-0.82999983 -0.90999992 -0.52999977 -0.5100001  -0.64000015 -0.77999994
  -0.90000023 -1.16000002 -1.0500001  -1.26       -1.08       -0.7600001 ]]
Average difference between predicted and actual: 0.8675000127156576
329/329 ━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step
Test RMSE: 1.9719857830020586
Test R^2: 0.9357155394620217
```

Model 5, 24 hour forecast given previous 48. (code says 12 hours, but it's really 24)

```
Predicted next 12 hours temperatures: [[5.68 6.04 6.36 6.67 6.9  7.08 7.16 7.21 7.15 7.   6.79 6.52 6.23 5.
  5.65 5.35 5.09 4.83 4.57 4.39 4.31 4.35 4.54 4.79]]
Actual next 12 hours temperatures : [5.94 6.27 6.87 7.41 7.36 7.67 7.49 7.48 7.4  7.33 6.79 6.61 6.55 6.47
 6.39 6.43 6.15 6.26 6.08 5.86 5.87 5.9  5.83 5.93]
Difference between predicted and actual: [[-2.60000172e-01 -2.30000038e-01 -5.09999866e-01 -7.39999924e-01
  -4.59999905e-01 -5.90000076e-01 -3.30000153e-01 -2.69999962e-01
  -2.49999905e-01 -3.30000000e-01 -3.81469727e-08 -9.00000191e-02
  -3.19999981e-01 -5.29999943e-01 -7.39999905e-01 -1.08000010e+00
  -1.05999985e+00 -1.43000008e+00 -1.50999983e+00 -1.47000013e+00
  -1.56000006e+00 -1.55000010e+00 -1.29000004e+00 -1.14000004e+00]]
Average difference between predicted and actual: 0.7391666706403098
329/329 ━━━━━━━━━━━━━━━━━━━━ 2s 6ms/step
Test RMSE: 2.3972543306993406
Test R^2: 0.9049434192256575
```

Key insights

All models demonstrated relatively high levels of model fitting, with none falling below 0.8. However, when comparing the Random Forest and LSTM models, the LSTM consistently achieved higher scores, with an R-squared value of approximately 0.90-0.96, even after tweaking hyperparameters such as the number of nodes and hidden layers. Increasing the window size for the inputs led to better predictive performances by providing a richer context for the sequential data.

However, there was minimal variability in accuracy across different configurations. Increasing the number of nodes and hidden layers only yielded slight improvements. This suggests that we can simplify the model's complexity without sacrificing predictive performance.

Based on these findings, Model 5 emerges as the most optimal choice, offering a balance between accuracy and computational efficiency. As well as being the most useful in forecasting many hours into the future.

While our results did not reveal any anomalies, there were notable limitations in our dataset that could impact the development of a more realistic and practical model. The variables within our data were relatively simplistic and did not fully capture the inherent complexities of weather forecasting, where numerous interconnected factors—such as wind patterns, ocean currents, and atmospheric pressure—play a critical role.

Furthermore, the data was sourced from Germany, a region characterized by relatively predictable seasonal patterns, with hot summers and cold winters. This stability in weather reduces the likelihood of significant anomalies, which means our model may not effectively account for unexpected deviations from the norm. Incorporating data from regions with more dynamic and unpredictable weather conditions could help the model better handle such irregularities, ultimately improving its generalizability and robustness.

---

5. Discussion

Interpretation of results

Hyperparameter Tuning: Training LSTMs is computationally intensive, and the marginal gains observed by tweaking hyperparameters suggest diminishing returns as model complexity increases. This required balancing the trade-off between model accuracy and computational overhead.

Generalization: The dataset's geographic specificity (Germany) and the relatively stable seasonal climate posed limitations for the model's broader applicability. Generalizing the model to regions with more dynamic weather systems could present additional challenges.

Scalability: As the forecasting window increases, accuracy slightly declines, indicating the need for architectures like Transformers to handle larger input and output sequences efficiently.

---

6. Conclusion and Future Work

In conclusion, this paper highlights the effectiveness of LSTM models in temperature forecasting, demonstrating their ability to capture temporal dependencies and deliver superior performance compared to traditional methods. By leveraging LSTM's capacity for learning sequential patterns, the models achieved high accuracy, providing a promising approach for predictive tasks in weather and climate domains. However, there is room for further improvement. Incorporating external factors such as CO_2 levels, sea surface temperatures, and other environmental variables could improve the model's ability to account for broader climatic influences. Expanding the scope to multi-variable and multi-location forecasting would enable the development of more globally applicable solutions. Additionally, utilizing other deep learning architectures, such as Transformers, may offer greater efficiency and predictive accuracy. Overall, this project can improve future forecasting models for a wide range of applications.

References

Libraries

Matplotlib.

Scikit-learn.

TensorFlow.

Keras.