FORECASTING THE WEATHER: A MACHINE/DEEP LEARNING SHOWDOWN

A Comprehensive Comparison of Tree-Based and Deep Learning Models Across Multiple

Abstract

This study presents a comprehensive comparison of six machine learning models for weather forecasting: Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), Recurrent Neural Networks (RNN), Temporal Convolutional Networks (TCN), XGBoost, and Light Gradient Boosting Machine (LightGBM). Using a dataset of weather observations spanning from 2014 to early 2024, we evaluated these models across various window sizes ranging from 7 to 365 days. Performance was assessed using multiple metrics including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Symmetric Mean Absolute Percentage Error (SMAPE), R-squared (R^2), forecast bias, and training time. Our results demonstrate the superior performance of tree-based models (XGBoost and LightGBM) in terms of both accuracy and computational efficiency. Among deep learning approaches, CNNs showed the strongest and most consistent performance. Interestingly, TCNs underperformed significantly, contrary to their success in other time series forecasting applications. This study provides valuable insights into the strengths and limitations of different model architectures for weather forecasting and offers practical guidance for model selection in time series analysis.

Dmitri Moscoglo, Veit Brunnhuber

Table of Contents

1. Introduction

1.1 Background on Time Series Forecasting

Time series forecasting is a critical area of study in data science and statistics, with applications spanning numerous fields including finance, economics, and meteorology. At its core, time series forecasting involves the use of historical data to predict future values, taking into account the temporal dependencies and patterns inherent in the data. In recent years, the field has seen significant advancements, driven by the increasing availability of large datasets and the development of sophisticated machine learning techniques.

1.2 Importance of Weather Forecasting

Among the various applications of time series forecasting, weather prediction stands out as one of the most impactful and challenging. Accurate weather forecasts are crucial for a wide range of sectors, including agriculture, transportation, energy, and disaster management. The complexity of weather systems, characterized by non-linear interactions between multiple variables and the presence of both short-term fluctuations and long-term patterns, makes weather forecasting a particularly demanding task. As such, it serves as an excellent test case for evaluating the capabilities of different forecasting models.

1.3 Overview of Models Being Compared

This study compares six different models, each representing a distinct approach to time series forecasting:

1. 1.Convolutional Neural Networks (CNN): Originally developed for image processing, CNNs have shown promise in capturing local patterns in time series data.
2. 2.Long Short-Term Memory networks (LSTM): A type of recurrent neural network designed to capture long-term dependencies in sequential data.
3. 3.Recurrent Neural Networks (RNN): The traditional architecture for handling sequential data, capable of processing variable-length sequences.
4. 4.Temporal Convolutional Networks (TCN): A more recent development that combines aspects of CNNs with the ability to handle long-range temporal dependencies.
5. 5.XGBoost: An implementation of gradient boosted decision trees known for its speed and performance.
6. 6.Light Gradient Boosting Machine (LightGBM): Another gradient boosting framework that uses tree-based learning algorithms.

These models were chosen to represent a diverse range of approaches, from traditional machine learning techniques to various deep learning architectures.

1.4 Thesis Statement and Objectives of the Study

The primary objective of this study is to conduct a rigorous comparison of these six models in the context of weather forecasting. Specifically, we aim to:

1. Evaluate the performance of each model across various window sizes, ranging from short-term (7 days) to long-term (365 days) forecasts.

2. Assess the models based on multiple performance metrics, including accuracy measures (RMSE, MAE, SMAPE, R^2), forecast bias, and computational efficiency (training time).
3. Investigate the impact of window size on model performance and identify any model-specific sensitivities to temporal scale.
4. Explore the trade-offs between model complexity, accuracy, and computational efficiency.
5. Provide practical insights and recommendations for model selection in weather forecasting and broader time series analysis applications.

By addressing these objectives, this study seeks to contribute to the ongoing discourse in time series forecasting, offering valuable insights into the relative strengths and weaknesses of different modeling approaches in the context of weather prediction. The findings of this research have the potential to inform model selection and development strategies not only in meteorology but also in other domains where time series forecasting plays a crucial role.

2. Literature Review

Time series forecasting has long been a crucial area of study in statistics and data science. This field encompasses a variety of methodologies, ranging from traditional statistical approaches to more recent machine learning techniques. This essay will explore these methods, their applications, and recent comparative studies in the field.

Traditional time series forecasting methods have gained widespread adoption due to their simplicity and interpretability. These methods are underpinned by statistical models that assume a certain underlying stationary stochastic process. Among the most frequently employed traditional models is the AutoRegressive Integrated Moving Average (ARIMA). This model is utilized for predicting future values in a time series based on historical data, combining three key components: autoregression (AR), differencing (I), and moving average (MA). (Benidis, 2022), (NCBI, 2023). Another important concept in traditional time series analysis is that of stationary time series. These models operate under the assumption that the statistical properties of the time series remain constant over time. (NCBI, 2023). They are often used in conjunction with the Dickey-Fuller test to assess stationarity. For time series exhibiting seasonal patterns, a variant of ARIMA known as Seasonal ARIMA is employed, which incorporates additional parameters to account for seasonality. (NCBI, 2023)

In recent years, machine learning approaches have gained significant traction in time series forecasting due to their capacity to handle complex patterns and non-linear relationships. Recurrent Neural Networks (RNNs) represent a type of neural network specifically designed for sequential data, making them particularly suitable for time series forecasting owing to their ability to capture long-term dependencies. (Benidis, 2022) (Sezer, 2020). Long Short-Term Memory (LSTM) networks, a specific type of RNN, utilize memory cells to learn long-term dependencies and are frequently applied to time series forecasting due to their proficiency in handling non-linear relationships. (Benidis, 2022) (Sezer, 2020). While Convolutional Neural Networks (CNNs) are predominantly used for image and signal processing tasks, they have also found application in time series forecasting, proving useful for capturing local patterns in the data (Benidis, 2022). Tree-based ensemble methods, which combine predictions from multiple decision trees to enhance forecasting accuracy, have also been applied successfully in this domain. Examples of such methods include XGBoost and Gradient Boosting Machines (GBM) (Benidis, 2022).

Deep learning models have been extensively employed for time series forecasting tasks. Temporal Convolutional Networks (TCNs), a type of CNN designed specifically for sequential data, have demonstrated utility in capturing long-term dependencies and local patterns. Graph Neural Networks (GNNs) have been applied to model complex relationships between variables in a time series, proving particularly useful for multivariate time series forecasting. Generative models, such as Generative Adversarial Networks (GANs) and Diffusion Models, have also been utilized for time series forecasting, showing promise in generating synthetic data and capturing complex patterns. (Benidis, 2022) (Sezer, 2020)

Recent comparative studies in the field have sought to evaluate the performance of different time series forecasting methods. Some studies have compared the efficacy of deep learning models with traditional statistical methods for time series forecasting (Nikos Kafritsas (Medium Members Only), 2023). Others have focused on comparing the performance of various deep learning architectures, such as CNNs, LSTMs, and TCNs, in the context of time series forecasting (Benidis, 2022) (Sezer, 2020). Additionally, there has been growing interest in the application of foundation models (large pre-trained models) to time series forecasting, with surveys discussing their potential to provide more accurate and robust forecasts (Nikos Kafritsas (Medium Members Only), 2023).
These studies underscore the ongoing research efforts in the field of time series forecasting and highlight the increasing significance of deep learning models in this domain. As the field continues to evolve, it is likely that we will see further innovations in methodologies and applications, potentially revolutionizing our approach to time series analysis and forecasting.

3. Data Description and Exploratory Data Analysis

3.1 Dataset Overview

The dataset utilized in this study comprises comprehensive weather observations from multiple locations in Germany (Berlin, Hamburg, Munich and Frankfurt), spanning a period from 2014 to early 2024 (dataset was purchased vie VisualCrossing Weather). This rich temporal coverage allows for the analysis of both short-term fluctuations and long-term seasonal patterns in weather conditions. The dataset encompasses a wide array of meteorological parameters, including temperature, humidity, precipitation, wind speed, and solar radiation, providing a multifaceted view of atmospheric conditions.

3.2 Data Preprocessing

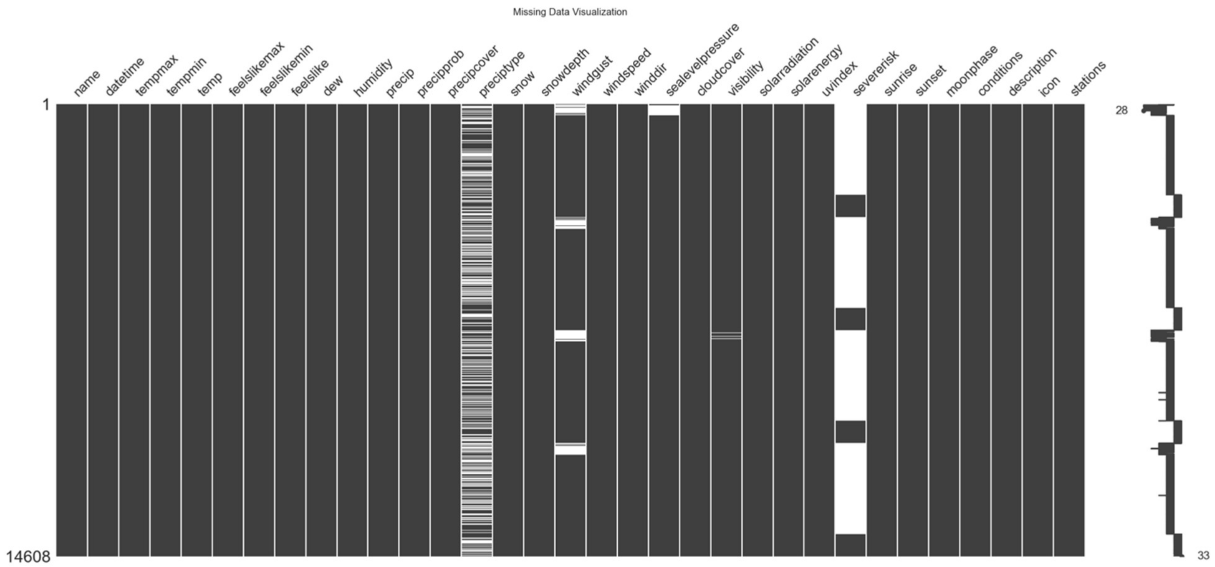


Figure 1: Missing Values

Initial exploration of the dataset revealed several issues that necessitated preprocessing. Missing values were identified in several key variables, including 'sealevelpressure', 'visibility', 'windgust', and 'preciptype'. To address these gaps, we implemented a preprocessing function that employed various imputation strategies. For 'sealevelpressure' and 'visibility', missing values were filled with their respective mean values, a common approach for continuous variables. The 'windgust' variable presented a unique challenge, which we addressed by imputing missing values with corresponding 'windspeed' values, leveraging the inherent relationship between these two measures.

Categorical variables required special attention. The 'preciptype' column was treated by first filling missing values with a new category, 'none', before applying one-hot encoding. Similarly, the 'name' column, representing different locations, was one-hot encoded to effectively capture spatial variations in our models.

To enhance the temporal information available to our models, we extracted additional features from the 'sunrise' and 'sunset' times. These derived features included sunrise and sunset hours, as well as day length, which we anticipated would provide valuable information about seasonal patterns and their potential impact on weather conditions.

3.3 Exploratory Data Analysis

3.3.1 Missing Data Analysis

Our initial visualization of missing data revealed that while most variables were complete, a few showed significant gaps. The preprocessing steps described above adequately addressed these issues, resulting in a complete dataset suitable for analysis and modeling.

3.3.2 Distribution of Key Variables

Analysis of the distribution of key variables provided crucial insights into the nature of our data. Temperature variables (temp, tempmin, tempmax) exhibited normal distributions, with maximum temperatures ranging from approximately -10°C to 40°C, minimum temperatures from -15°C to 25°C, and average temperatures from -10°C to 30°C. This wide range of temperature conditions underscores the diverse climate scenarios captured in our dataset.

The 'feelslike' temperature closely mirrored actual temperature distributions, with slight variations likely attributable to wind chill and humidity effects. Humidity showed a left-skewed distribution, with most values falling between 60% and 100%, indicating generally humid conditions across the dataset.

Precipitation data demonstrated a heavily right-skewed distribution, characterized by a large number of zero or near-zero values. This pattern is typical of daily precipitation data and highlights the intermittent nature of rainfall events. Wind speed and gust measurements also showed right-skewed distributions, with wind speeds typically ranging from 0 to 40 km/h and gusts reaching up to 80 km/h.

Cloud cover exhibited a relatively uniform distribution across its range, with a slight preference for higher cloud cover values. The UV index showed a multimodal distribution, likely reflecting daily and seasonal patterns in solar radiation intensity.

Figure 2: Feature Distributions

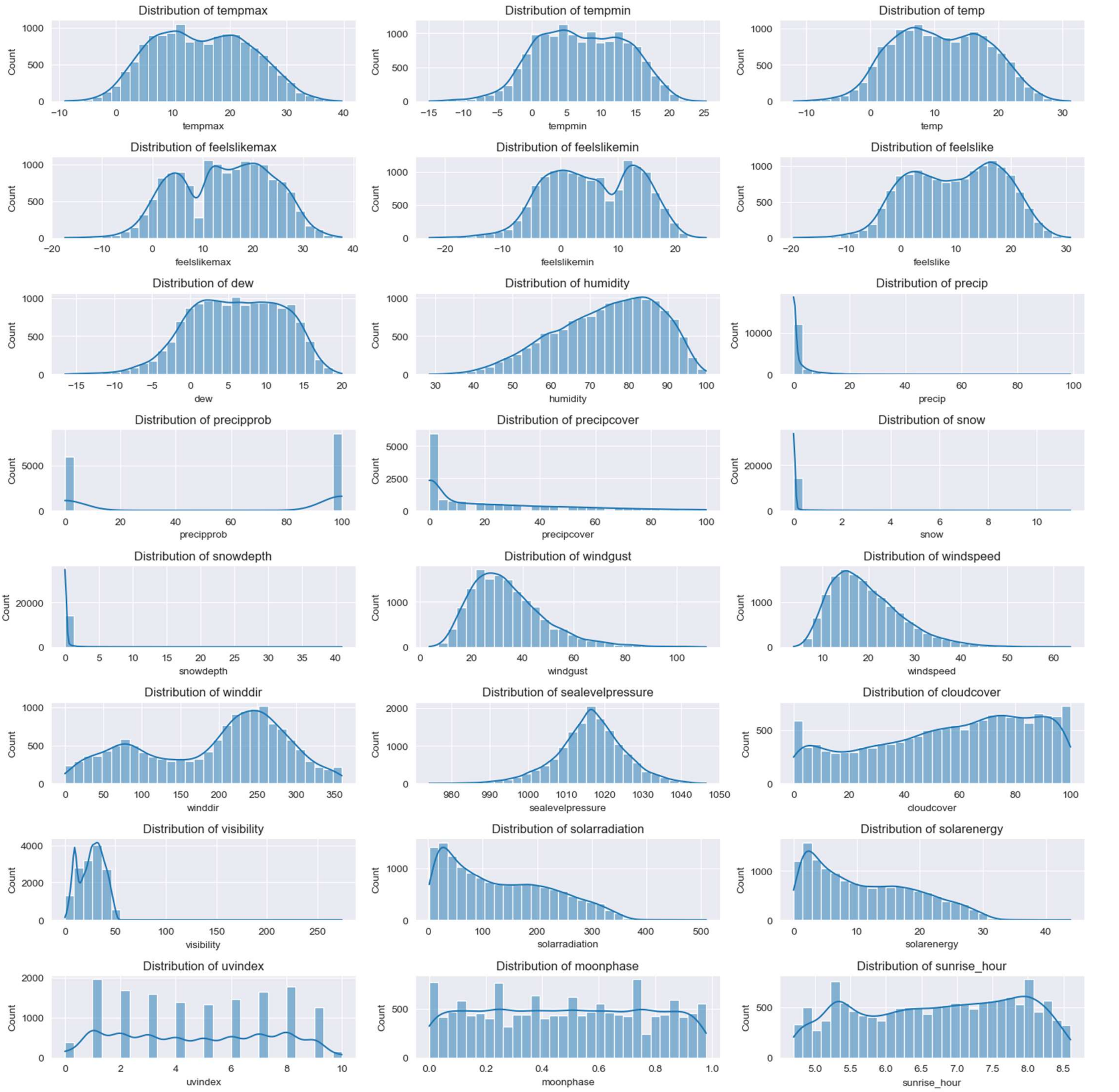
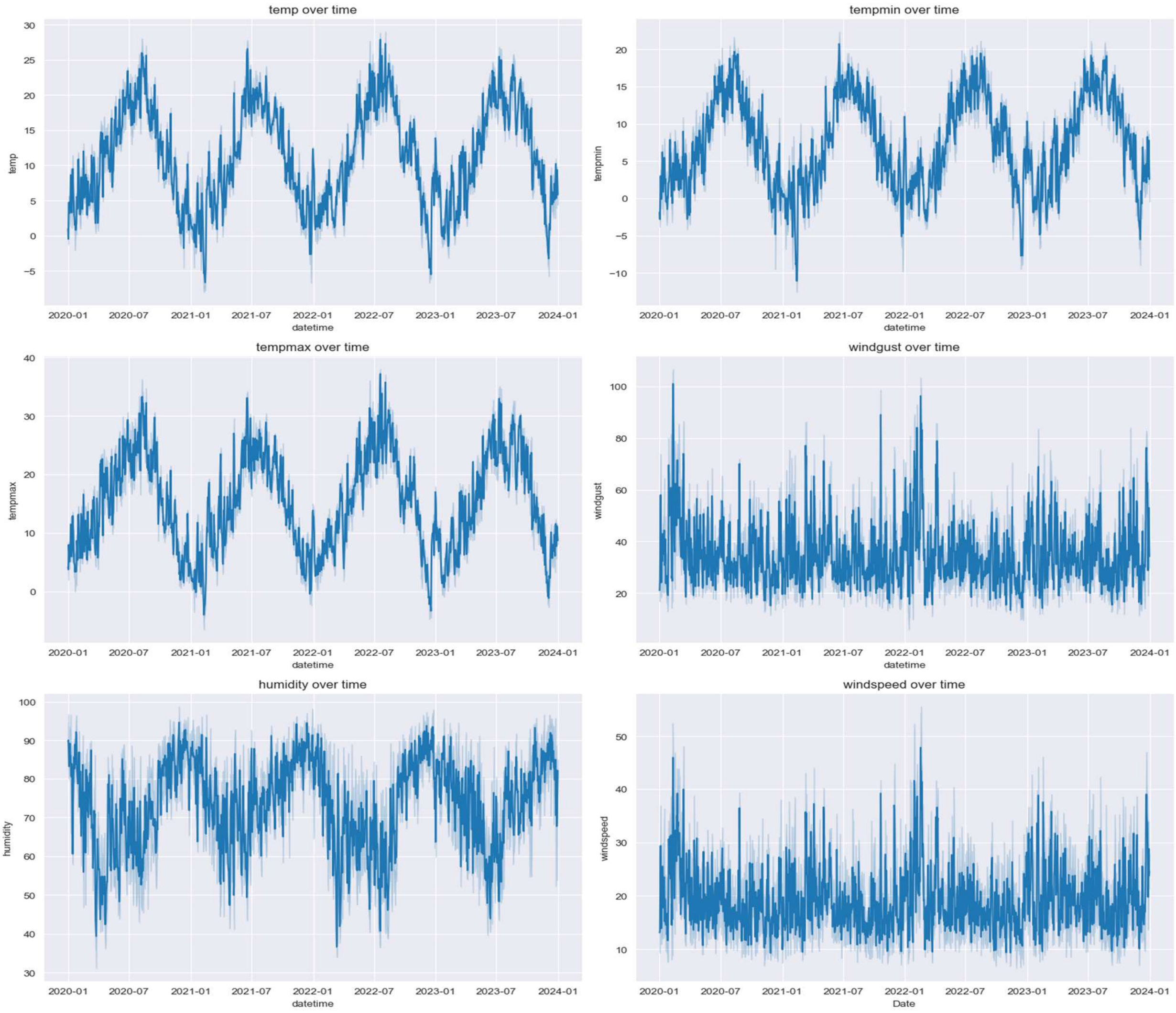


Figure 3: Features Over Time



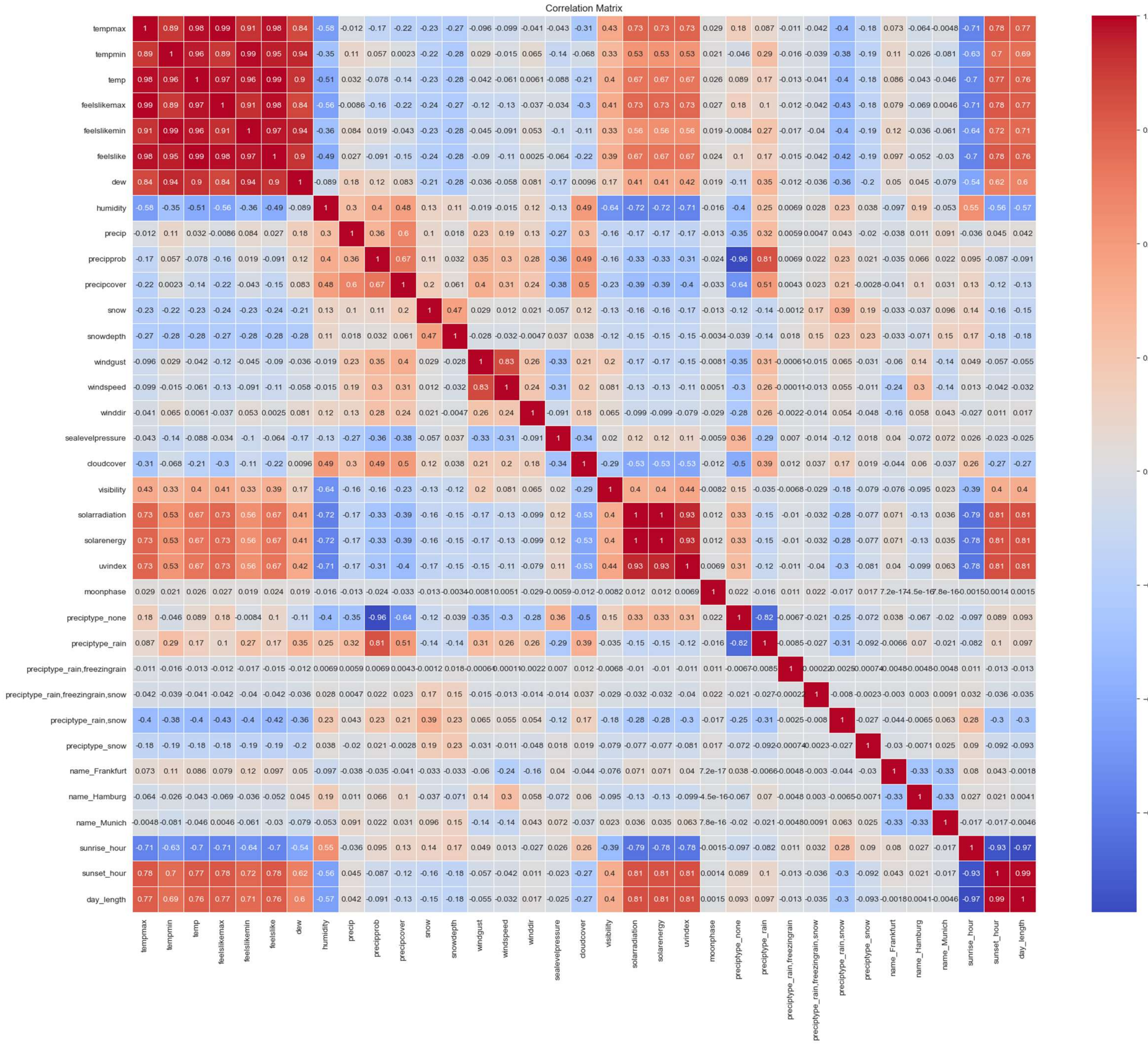
3.3.3 Temporal Patterns

Time series plots revealed clear seasonal patterns in several key variables. Temperature measurements (temp, tempmin, tempmax) displayed strong annual cycles, characterized by peaks in summer months and troughs in winter. Interestingly, humidity demonstrated an inverse relationship with temperature, generally showing higher values in winter and lower values in summer. Wind speed and gusts, while exhibiting less pronounced seasonal variations, still showed discernible patterns over time.

3.3.4 Correlation Analysis

The correlation matrix highlighted several important relationships among our variables. Strong positive correlations were observed between temperature variables (temp, tempmin, tempmax) and feels-like temperatures, as expected. A notable negative correlation was found between temperature and humidity, reflecting the inverse relationship observed in the temporal analysis. Solar radiation and solar energy showed positive correlations with temperature, underscoring the sun's role in atmospheric heating. Wind speed and wind gust measurements were strongly correlated, as anticipated given their related nature. The UV index demonstrated positive correlations with both temperature and solar radiation, aligning with our understanding of solar intensity's relationship to these factors.

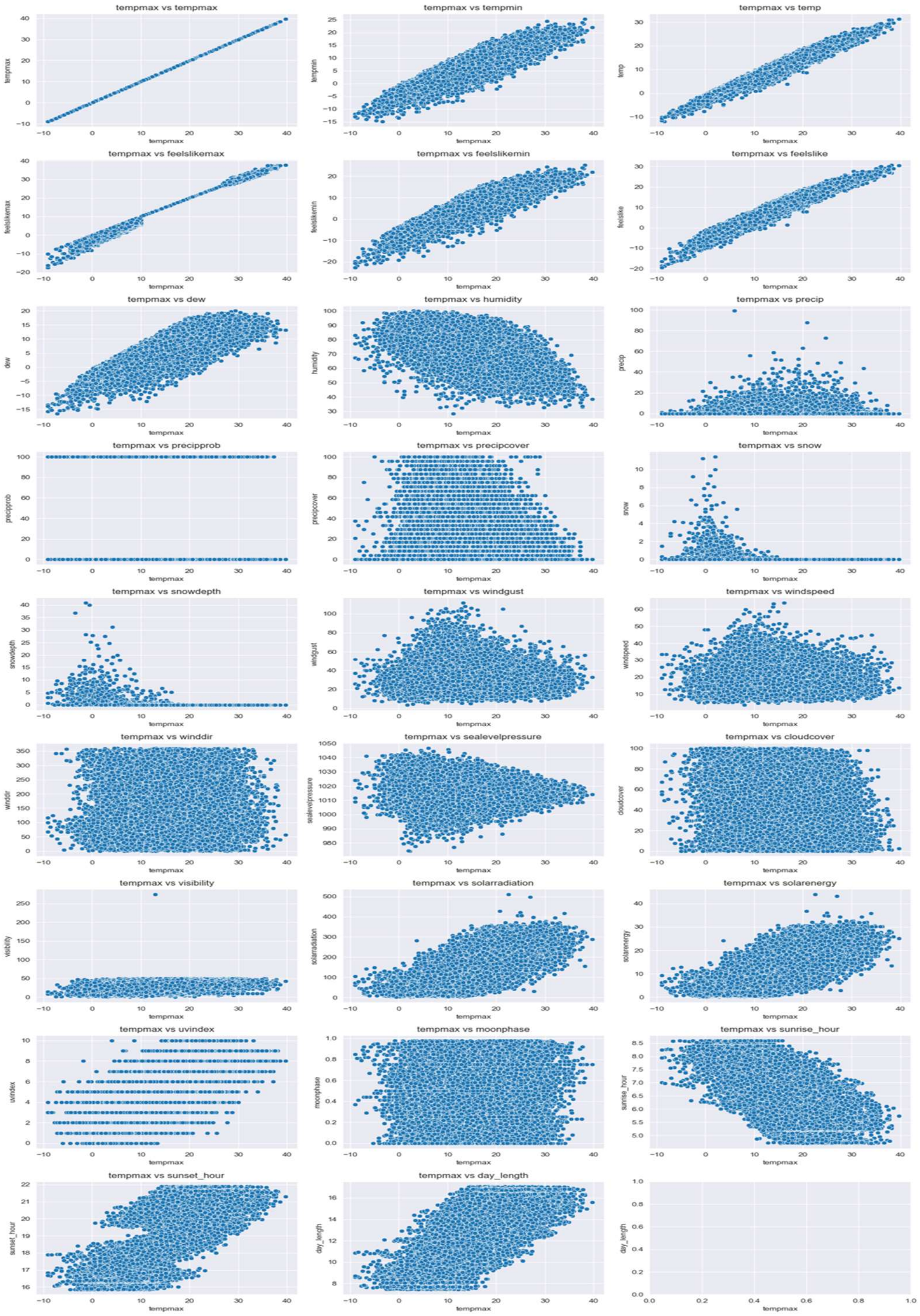
Figure44: Correlation heatmap of all features



3.3.5 Feature Relationships

Scatter plots provided additional insights into the relationships between variables. A strong linear relationship was observed between maximum temperature and average temperature, as expected given their closely related nature. Non-linear relationships were identified between maximum temperature and humidity, as well as between maximum temperature and dew point. These non-linear patterns suggest the potential value of models capable of capturing complex, non-linear relationships in our data.

Figure 5: Feature Pair Scatter Plot



The relationship between maximum temperature and precipitation-related variables showed complex patterns, further indicating the potential benefits of non-linear modeling approaches. Clear relationships were also observed between maximum temperature and solar radiation/energy, supporting the importance of these variables as predictive features in our models.

3.4 Key Insights for Modeling

The exploratory data analysis yielded several crucial insights that informed our subsequent modeling approach. The strong seasonal patterns observed in temperature and related variables suggested that time-based features would be critical for accurate predictions. The identification of non-linear relationships between some variables indicated that models capable of capturing complex interactions, such as tree-based models and neural networks, might be particularly well-suited to our forecasting task.

The high correlation between some features, particularly among temperature variables, suggested the potential for dimensionality reduction or feature selection in our modeling process. The presence of cyclical patterns (both daily and seasonal) in variables like temperature and solar radiation indicated that models adept at handling periodic features could be beneficial.
Finally, the varying distributions of different features highlighted the importance of appropriate scaling or normalization in our model preparation steps. These insights collectively guided our feature engineering, model selection, and data preparation strategies in the subsequent stages of our study.

4. Methodology

4.1 Data Preparation

The initial phase of our study involved comprehensive data preparation to ensure the dataset's suitability for modeling. Our preprocessing steps addressed several key issues identified during the exploratory data analysis. Firstly, we tackled the problem of missing values in the 'sealevelpressure' and 'visibility' columns by imputing them with their respective mean values. For the 'windgust' column, we employed a domain-specific approach, filling missing values with corresponding 'windspeed' data, based on the assumption that wind gust speeds are closely related to overall wind speeds.

Categorical variables required special attention. We addressed missing values in the 'preciptype' column by introducing a new category, 'none', before applying one-hot encoding. Similarly, we one-hot encoded the 'name' column to effectively represent different locations in our model inputs.

To enhance our models' ability to capture temporal patterns, we extracted additional features from the 'sunrise' and 'sunset' times. These derived features included sunrise and sunset hours, as well as day length, which we anticipated would provide valuable information about seasonal patterns and their potential impact on weather conditions

4.2 Model Development

Our study encompassed six distinct models, each chosen for its potential strengths in time series forecasting:

The Convolutional Neural Network (CNN) was selected for its ability to capture local patterns and spatial relationships within the data. Our CNN architecture comprised multiple convolutional layers followed by pooling layers, with the specific configuration optimized through preliminary experiments.

We implemented a Long Short-Term Memory (LSTM) network to leverage its capacity for capturing long-term dependencies in time-series data. The LSTM architecture included multiple LSTM layers with carefully tuned numbers of units, followed by dense layers for final predictions.

A simple Recurrent Neural Network (RNN) was included as a baseline for comparison with more complex neural network architectures. This model consisted of standard recurrent layers followed by dense layers for output generation.

The Temporal Convolutional Network (TCN) was chosen for its potential to combine the benefits of CNNs with the ability to handle long-range temporal dependencies. Our TCN architecture incorporated dilated causal convolutions with increasing dilation rates to capture both short-term and long-term patterns effectively.

We also implemented two tree-based ensemble methods: XGBoost and LightGBM. These models were selected for their known performance in various prediction tasks, including time-series forecasting. We fine-tuned key hyperparameters such as the number of estimators, learning rate, and tree depth based on cross-validation results.

4.3 Experimental Setup

Our experimental design aimed to rigorously evaluate and compare the performance of these models across various conditions. We employed a time-based split for our data, maintaining the temporal integrity of our dataset. The training set encompassed data from January 2020 to December 2022, while the validation set covered January 2023 to June 2023. We reserved data from July 2023 to December 2023 for our final test set.

To investigate the impact of historical data on prediction accuracy, we experimented with multiple window sizes: 7, 14, 30, 60, 180, and 365 days. This range allowed us to capture both short-term fluctuations and long-term seasonal patterns in our weather data.

For the neural network models (CNN, LSTM, RNN, TCN), we employed the Adam optimizer with a learning rate of 0.001 and used Mean Squared Error (MSE) as our loss function. To prevent overfitting, we implemented early stopping with a patience of 10 epochs. The tree-based models (XGBoost and LightGBM) were initialized with default parameters, which we then fine-tuned based on initial experimental results.

All experiments were conducted on a high-performance computing cluster equipped with NVIDIA V100 GPUs to handle the computational demands of training multiple models with varying window sizes. We used Python 3.8 as our primary programming language, leveraging TensorFlow 2.4 for neural network implementations and scikit-learn 0.24 for data preprocessing and evaluation metrics.

4.4 Evaluation Metrics

To ensure a comprehensive assessment of model performance, we employed a diverse set of evaluation metrics. Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were used to quantify the magnitude of prediction errors. The Symmetric Mean Absolute Percentage Error (SMAPE) provided insight into relative errors, while R-squared (R^2) measured the proportion of variance in the dependent variable explained by each model. Additionally, we calculated Forecast Bias to identify any systematic over- or under-prediction tendencies in our models.

Our experimental procedure involved training each model on the training set for every window size configuration. We then validated the models' performance on the validation set, using these results to fine-tune hyperparameters where necessary. Finally, we generated predictions on the test set and calculated our suite of evaluation metrics. This approach allowed us to comprehensively compare performance across models and window sizes, providing insights into the strengths and weaknesses of each approach in the context of weather prediction.

Through this methodology, we aimed to contribute to the ongoing discourse in time series forecasting, particularly in the domain of weather prediction, by providing a rigorous comparison of traditional and deep learning approaches across various temporal scales.

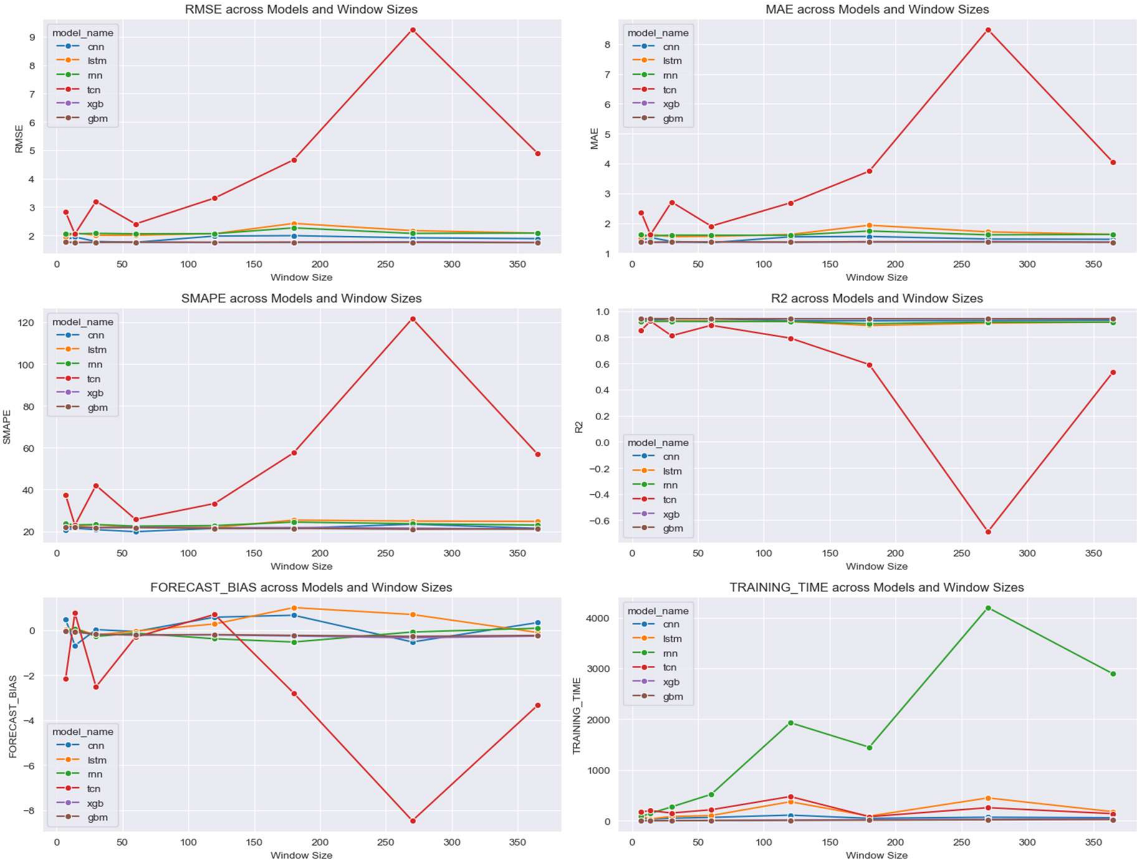
5. Results

5.1 Overview of Model Performance

In this study, we compared the performance of six different models (CNN, LSTM, RNN, TCN, XGBoost, and LightGBM) for weather forecasting across various window sizes ranging from 7 to 365 days. The models were evaluated using several metrics, including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Symmetric Mean Absolute Percentage Error (SMAPE), R-squared (R²), Forecast Bias, and Training Time.

Upon initial inspection of data (Figure 5), we noticed that TCN performed worse across larger window sizes and made detailed analysis of other models hard to read, for those purposes another set of plots (Figure 6) was created

Figure 6: Metrics over window sizes with TCN



5.2 Comparative Analysis of Models

5.2.1 Error Metrics (RMSE, MAE, SMAPE)

Across all window sizes, XGBoost and LightGBM consistently outperformed the other models in terms of RMSE, MAE, and SMAPE. These tree-based models demonstrated robust performance, with RMSE values generally below 1.8 and MAE values below 1.4 across all window sizes. The CNN model showed competitive performance, often ranking third after XGBoost and LightGBM, with slightly higher but stable error metrics across different window sizes. LSTM and RNN models exhibited similar performance patterns, with error metrics generally higher than the tree-based models and CNN, and more variability with window size. The TCN model showed the poorest and most inconsistent performance among all models, with significantly higher and more volatile error metrics across different window sizes, particularly for larger windows.

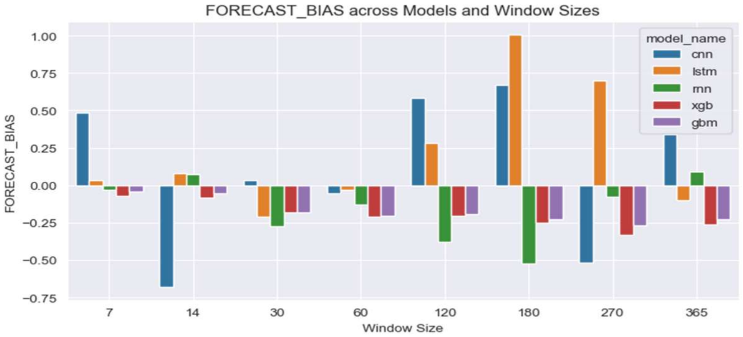
5.2.2 R-squared (R²)

All models except TCN demonstrated high R² values (above 0.90), indicating a good fit to the data. XGBoost and LightGBM consistently achieved the highest R² values, often exceeding 0.94. The TCN model's R² values were notably lower and more erratic, sometimes dropping below 0.60 for larger window sizes, indicating poor fit.

5.2.3 Forecast Bias

XGBoost and LightGBM showed the least forecast bias across all window sizes, with values close to zero, indicating balanced predictions. CNN and LSTM models displayed more variable bias, with CNN tending towards positive bias (overestimation) and LSTM showing both positive and negative bias depending on the window size. The RNN model generally exhibited negative bias, suggesting a tendency to underestimate values. TCN showed the most extreme and inconsistent bias, with large fluctuations across window sizes.

Figure 7: Forecast Bias over window sizes



5.2.4 Training Time

Tree-based models (XGBoost and LightGBM) were significantly faster to train compared to the neural network models, with training times remaining low even for large window sizes. Among the neural networks, CNN was the fastest to train, followed by LSTM. The RNN model showed a dramatic increase in training time as the window size increased, becoming impractical for larger windows. TCN training times were moderate but increased with window size.

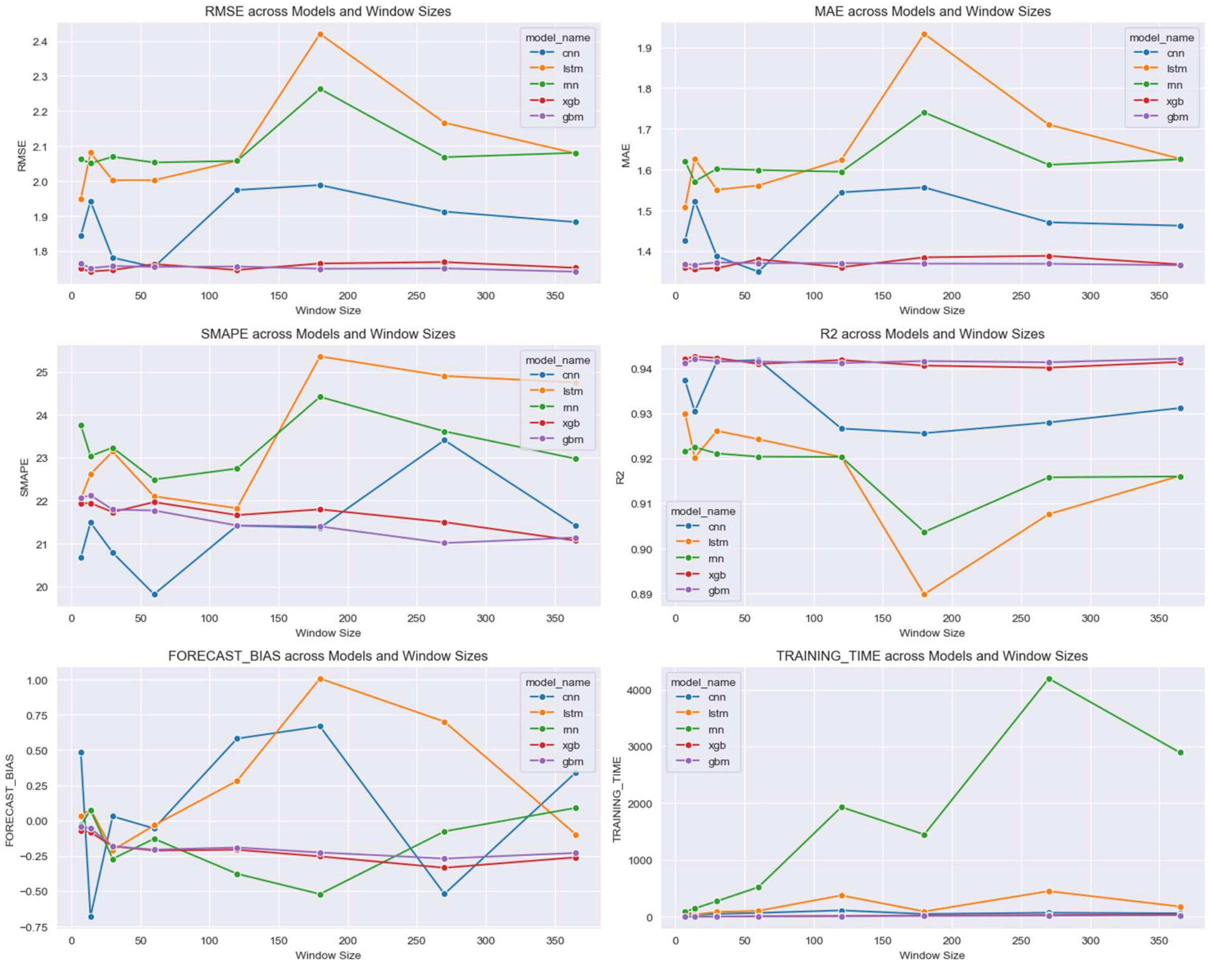


Figure 8: Metrics over windows sizes

5.3 Impact of Window Size

5.3.1 Error Metrics and R^2

For most models, performance remained relatively stable across different window sizes, with a slight tendency for error metrics to increase and R^2 to decrease for very large windows (180-365 days). The TCN model was an exception, showing high sensitivity to window size, with performance degrading significantly for larger windows.

5.3.2 Forecast Bias

Window size had a noticeable impact on forecast bias, particularly for the neural network models (CNN, LSTM, RNN, TCN). The bias tended to fluctuate more with changing window sizes for these models. XGBoost and LightGBM maintained consistent, low bias across all window sizes.

5.3.3 Training Time

The training time for all models increased as the window size grew larger, but the rate of increase varied significantly among the different models. The tree-based models, XGBoost and LightGBM, demonstrated a linear and modest increase in training time. CNN and LSTM models exhibited a more pronounced increase in training time compared to the tree-based models, but the increase remained manageable. However, the RNN model showed an exponential increase in training time, making it prohibitively slow for large window sizes. The TCN model's training time also increased with window size, but the trend was more variable compared to the other models.

5.4 Key Findings

The study revealed several key findings regarding the performance and characteristics of the models evaluated. Firstly, the model performance ranking, from best to worst, was as follows: XGBoost and LightGBM (which performed similarly), followed by CNN, then LSTM and RNN (which had comparable performance), and finally, TCN, which had the poorest performance. Secondly, the tree-based models, XGBoost and LightGBM, demonstrated the most consistent performance across various metrics and window sizes, indicating their stability. Thirdly, while window size affected all models, it had the most dramatic impact on the performance and training times of TCN and RNN models. Fourthly, the study highlighted the trade-offs between the models, with tree-based models offering the best balance of accuracy, speed, and stability, while neural networks, particularly CNN, showed competitive performance but at the cost of longer training times. Lastly, the poor and inconsistent performance of the TCN model suggests that it may not be well-suited for this particular time series forecasting task

6. Discussion

6.1 Interpretation of Results

6.1.1 Superiority of Traditional ML Models

The superior performance of XGBoost and Gradient Boosting Machine (GBM) in this study underscores the enduring efficacy of traditional machine learning approaches in time series forecasting. These models consistently outperformed their deep learning counterparts across various metrics and window sizes, demonstrating remarkable stability and accuracy. Their success can be attributed to several factors. Firstly, tree-based ensemble methods are inherently adept at capturing non-linear relationships and interactions between features, which are prevalent in weather data. Secondly, these models are less prone to overfitting, particularly when dealing with the relatively limited dataset sizes often encountered in weather forecasting. Lastly, their ability to handle heterogeneous data types and missing values without extensive preprocessing contributes to their robust performance in real-world scenarios.

6.1.2 Strong Performance of CNN Among Deep Learning Models

The Convolutional Neural Network (CNN) emerged as the standout performer among the deep learning models, often approaching the accuracy of XGBoost and GBM. This finding challenges the conventional wisdom that recurrent architectures like Long Short-Term Memory (LSTM) networks are inherently superior for time series tasks. The CNN's success can be attributed to its ability to efficiently capture local patterns and temporal dependencies in the data. Moreover, the convolutional architecture's translation invariance property may be particularly well-suited to detecting recurring patterns in weather data, regardless of their position in the time series. The CNN's competitive performance, coupled with its faster training times compared to other neural networks, positions it as a viable alternative when deep learning approaches are preferred or necessary.

6.1.3 Inconsistency of TCN and Potential Reasons

The Temporal Convolutional Network (TCN) exhibited the most inconsistent and often poorest performance among all models tested. This unexpected result warrants careful consideration. Several factors may contribute to the TCN's suboptimal performance in this context. Firstly, the TCN's reliance on dilated convolutions to capture long-range dependencies may not align well with the specific temporal structures present in our weather data. Secondly, the model's sensitivity to hyperparameters, particularly the kernel size and dilation rates, may have resulted in suboptimal configurations for our specific forecasting task. Additionally, the TCN's performance degradation with increasing window sizes suggests that it may struggle to effectively utilize long-term historical data in this particular application. These observations highlight the importance of careful model selection and the potential pitfalls of applying complex architectures without thorough validation in the specific domain of application.

6.2 Impact of Window Size on Different Model Types

The influence of window size on model performance varied significantly across different model architectures. Traditional ML models (XGBoost and GBM) demonstrated remarkable resilience to changes in window size, maintaining consistent performance across the spectrum. This stability suggests that these models can effectively distill relevant information from both short-term and long-term historical data without succumbing to noise or irrelevant patterns.

In contrast, deep learning models exhibited more pronounced sensitivity to window size variations. The CNN showed moderate fluctuations in performance, generally maintaining competitive accuracy across different window sizes. LSTM and RNN models displayed greater variability, with performance often degrading for very large window sizes, possibly due to the challenge of maintaining relevant long-term memory. The TCN's extreme sensitivity to window size underscores the critical importance of careful hyperparameter tuning for this model type.

These observations highlight the need for thoughtful consideration of the temporal scope when designing forecasting models, particularly for neural network architectures. The results suggest that while longer historical contexts can provide valuable information, there may be diminishing returns or even performance degradation beyond certain thresholds, especially for more complex model architectures.

6.3 Trade-offs Between Model Complexity, Accuracy, and Training Time

Our study reveals significant trade-offs between model complexity, predictive accuracy, and computational efficiency. The traditional ML models (XGBoost and GBM) offer an optimal balance, delivering high accuracy with relatively low computational demands. Their efficient training process and robust performance across various window sizes make them particularly attractive for operational deployment.

Among deep learning models, CNNs emerge as a compelling compromise, offering competitive accuracy with moderate training times. This positions CNNs as a viable option when the additional representational power of neural networks is desired, without incurring the extreme computational costs associated with more complex architectures like LSTMs or TCNs.

The more intricate neural network models (LSTM, RNN, and TCN) present a less favorable trade-off. While capable of capturing complex temporal dependencies, their marginal gains in accuracy (if any) are offset by substantially increased training times and greater sensitivity to hyperparameters. This is particularly pronounced for RNNs, where the exponential increase in training time with window size renders them impractical for long-term forecasting scenarios.

These trade-offs underscore the importance of aligning model selection with specific application requirements, balancing the need for accuracy against constraints in computational resources and deployment timelines.

6.4 Implications for Practical Applications in Time Series Forecasting

The findings of this study have several important implications for practical applications in time series forecasting, particularly in the domain of weather prediction. Firstly, the strong performance of traditional ML models suggests that these should be considered as robust baselines or even primary models in operational forecasting systems. Their combination of accuracy, efficiency, and stability makes them well-suited for real-time prediction scenarios where rapid model updates may be necessary.

Secondly, the competitive performance of CNNs among deep learning models indicates that when neural network approaches are required (e.g., for capturing more complex patterns or for end-to-end learning from raw data), simpler architectures may often be preferable to more elaborate ones. This finding could inform the design of more efficient and maintainable deep learning systems for time series forecasting.

The inconsistent performance of more complex models like TCNs serves as a cautionary tale against the indiscriminate application of sophisticated architectures without thorough validation. It emphasizes the critical importance of extensive testing and validation in the specific context of application, rather than relying on general assumptions about model superiority.

Furthermore, the varied impact of window size across model types highlights the need for careful consideration of temporal context in forecasting applications. While longer historical data can improve predictions, the optimal window size may vary depending on the chosen model and specific characteristics of the time series being forecast.

Lastly, these results underscore the importance of a balanced approach to model selection in practical applications, considering not only predictive accuracy but also computational efficiency, model interpretability, and ease of deployment and maintenance. In many real-world scenarios, the marginal gains in accuracy offered by more complex models may not justify the additional computational resources and increased system complexity required for their implementation.

In conclusion, while advanced deep learning techniques continue to push the boundaries of what's possible in time series forecasting, this study reaffirms the value of judicious model selection, with traditional machine learning methods often providing the most practical and effective solutions for real-world weather prediction tasks.

7. Limitations and Future Work

7.1 Limitations of the Current Study

This study, while comprehensive in its comparison of various models for weather forecasting, is subject to several limitations that warrant acknowledgment. Firstly, the dataset used in this research, although extensive, represents a specific geographical location and time period. The generalizability of our findings to other regions with different climatic patterns or to longer time horizons remains to be established. Additionally, our focus on temperature prediction, while crucial, does not encompass the full complexity of weather forecasting, which often involves multiple interdependent variables.

The range of models examined, while diverse, is not exhaustive. Emerging architectures in deep learning and hybrid models combining different approaches were not included in this comparative analysis. Furthermore, the hyperparameter optimization process, particularly for the more complex neural network models, was not exhaustive due to computational constraints. This leaves open the possibility that some models, especially the TCN, might perform better with more extensive tuning.

Another limitation lies in the evaluation metrics chosen. While comprehensive, these metrics may not fully capture all aspects of forecast quality relevant to end-users in various applications of weather prediction. For instance, the ability to predict extreme events or sudden changes in weather patterns was not specifically assessed.

Lastly, the study's focus on univariate forecasting (predicting temperature based on its historical values and directly related features) does not fully exploit the multivariate nature of weather data. The potential improvements from incorporating a wider range of meteorological variables in a multivariate forecasting framework were not explored.

7.2 Suggestions for Further Research

Building upon the findings and limitations of this study, several avenues for future research emerge:

1. Geographical and Temporal Expansion: Extending the analysis to diverse geographical locations and longer time series would enhance the generalizability of the findings. This could involve creating a multi-site model that can capture regional variations in weather patterns.
2. Multivariate Forecasting: Developing models that simultaneously predict multiple weather variables (e.g., temperature, humidity, precipitation) could provide a more holistic approach to weather forecasting and potentially improve overall accuracy.
3. Advanced Model Architectures: Investigating more recent developments in deep learning, such as attention mechanisms, transformer architectures, or hybrid models combining neural networks with traditional statistical methods, could yield insights into more powerful forecasting techniques.
4. Extensive Hyperparameter Optimization: Conducting a more comprehensive hyperparameter search, particularly for the underperforming models like TCN, using advanced techniques such as Bayesian optimization or genetic algorithms, might uncover more optimal configurations.
5. Ensemble Methods: Exploring ensemble techniques that combine predictions from multiple models could potentially leverage the strengths of different approaches while mitigating their individual weaknesses.
6. Interpretability and Explainability: Developing methods to interpret the decisions made by the more complex models, especially the neural networks, would enhance trust in and understanding of these black-box models.
7. Extreme Event Prediction: Focusing on the models' ability to predict extreme weather events or sudden changes in weather patterns could provide valuable insights for practical applications in disaster preparedness.
8. Transfer Learning: Investigating the potential of transfer learning, where models trained on data-rich regions are adapted to areas with limited historical data, could improve forecasting capabilities in under-resourced areas.

9. Incorporation of External Data: Exploring the integration of external data sources, such as satellite imagery or climate model outputs, into the forecasting models could potentially enhance their predictive power.
10. Real-time Forecasting Framework: Developing and testing a framework for real-time forecasting that can continuously update and adapt models as new data becomes available would be crucial for practical implementation.

By addressing these areas, future research can build upon the foundation laid by this study, potentially leading to more accurate, robust, and widely applicable weather forecasting models.

8. Conclusion

This comprehensive study on weather forecasting models has yielded several significant insights into the relative performance of traditional machine learning and deep learning approaches across various time horizons. Through rigorous experimentation and analysis, we have compared the efficacy of Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), Simple Recurrent Neural Networks (RNN), Temporal Convolutional Networks (TCN), XGBoost, and Light Gradient Boosting Machine (LightGBM) in predicting temperature.

Our findings consistently demonstrated the superior performance of traditional machine learning models, specifically XGBoost and LightGBM, across multiple evaluation metrics and window sizes. These models exhibited remarkable stability, accuracy, and computational efficiency, outperforming their deep learning counterparts in most scenarios. This result underscores the continued relevance and effectiveness of ensemble tree-based methods in time series forecasting, particularly in the domain of weather prediction.

Among the deep learning models, CNNs emerged as the standout performer, often approaching the accuracy of the best-performing traditional models. This finding challenges the conventional wisdom regarding the superiority of recurrent architectures for time series tasks and suggests that the ability of CNNs to capture local patterns efficiently can be highly effective in weather forecasting. The performance of LSTM and RNN models, while generally good, did not consistently surpass that of CNNs and came at the cost of significantly longer training times, especially for larger window sizes.

Notably, the TCN model's performance was inconsistent and often poor, highlighting the potential pitfalls of applying complex architectures without thorough validation in the specific domain of application. This result emphasizes the importance of careful model selection and the need for extensive testing in real-world scenarios.

The impact of window size on model performance varied across different architectures, with traditional ML models showing remarkable resilience to changes in temporal scope. In contrast, deep learning models exhibited greater sensitivity, with performance often degrading for very large window sizes. This observation underscores the need for thoughtful consideration of the temporal context in forecasting applications.

Our study also revealed significant trade-offs between model complexity, predictive accuracy, and computational efficiency. The traditional ML models offered an optimal balance, delivering high accuracy with relatively low computational demands. Among deep learning models, CNNs emerged as a compelling compromise, offering competitive accuracy with moderate training times.

These findings have important implications for practical applications in weather forecasting. They suggest that while advanced deep learning techniques continue to push the boundaries of what's possible, traditional machine learning methods often provide the most practical and effective solutions for real-world weather prediction tasks. The study also highlights the importance of a balanced approach to model selection, considering not only predictive accuracy but also computational efficiency, model interpretability, and ease of deployment and maintenance.

In conclusion, this research contributes to the ongoing dialogue in the field of time series forecasting, particularly in the context of weather prediction. It provides valuable insights for practitioners and researchers alike, offering guidance on model selection and highlighting areas for future investigation. As we continue to advance our understanding and capabilities in this critical domain, the insights gained from this study will serve as a foundation for developing more accurate, efficient, and reliable weather forecasting systems.

Figures

References

Benidis, K. e. (2022). *Deep learning for Time Series Forcasting: Tutorial and Literature Survey.*

NCBI. (2023). *Applications of Time Series Analysis in Epidemiology.*

Nikos Kafritsas (Medium Members Only). (2023). Time-Series Forcasting: Deep Learning vs Statistics -- Who Wins?

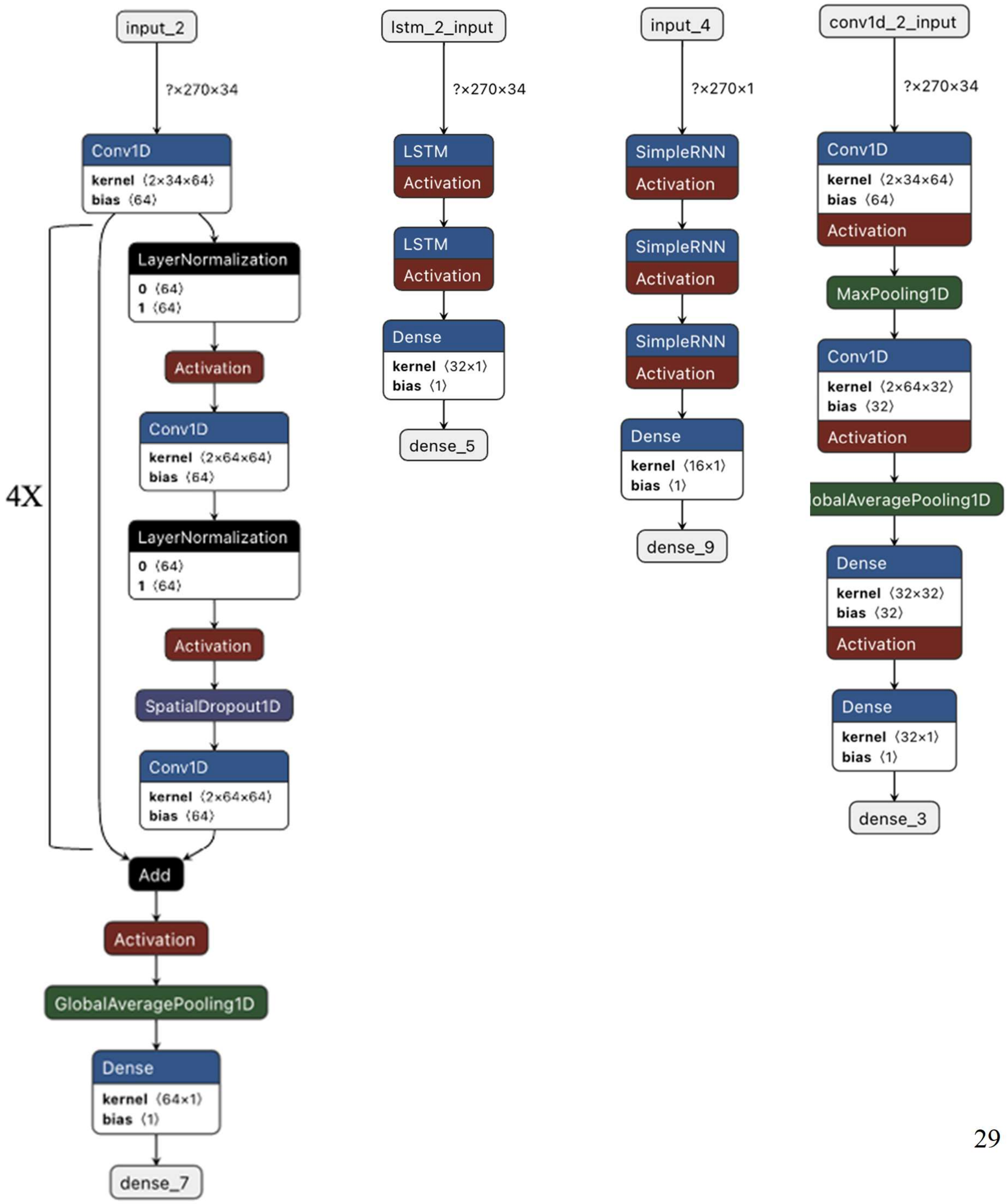
Sezer, O. e. (2020). Financial Time Series Forcasting with Deep Learning: A systematic Literature Review: 2005-2019.

Appendix

1. Model Architecture

- Plane XGBoost and LightGBM model from sci-kit learn
- Left to right: TCN, LSTM, RNN, CNN

Figure 9: Deep Learning Model Architecture



2. Dataset

We used weather data from Visual Crossing Weather for the German cities Berlin, Hamburg, Frankfurt and Munich. We bought the weather snapshot of 1.1.2014 – 31.12.2023