# **Hybrid Machine Learning Models in Weather Forecasting: A Comparative Study**

## **1. Background Research**

Weather forecasting traditionally relies on numerical models that simulate atmospheric physics. Prominent global Numerical Weather Prediction (NWP) systems like the European Centre’s IFS (ECMWF) and NOAA’s GFS solve complex fluid dynamics equations to predict future states of the atmosphere

[1](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=Let%E2%80%99s%20start%20with%20the%20%E2%80%98traditional%E2%80%99,surface%20to%20the%20upper%20atmosphere)

[2](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=Some%20of%20the%20most%20well,global%20deterministic%20NWP%20models%20are)

. These physics-based models ingest current observations and integrate the equations of motion forward in time on supercomputers, outputting forecasts of numerous weather variables (temperature, wind, pressure, etc.). Such models have been refined over decades and are the backbone of operational forecasts worldwide. For example, ECMWF’s model is often regarded as among the most accurate global forecast systems

[3](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=There%20is%20a%20consensus%20that,for%20all%20local%20NWP%20models)

. However, NWP models are computationally intensive and cannot easily learn from past errors because their design is based on fixed physical equations.

In recent years, there has been a surge of interest in applying Machine Learning (ML) to meteorology. Improvements in data availability (e.g. extensive reanalysis datasets) and computing power have enabled ML models to learn complex patterns from historical weather data. Notably, 2023 saw major developments in AI for weather forecasting: for instance, ECMWF announced an AI-based forecast system (AIFS) that emulates its physical model

[4](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=After%202023%20turned%20out%20to,potential%20benefits%20are%20sometimes%20overstated)

. These advances signal that data-driven models are beginning to play a central role in weather prediction

[5](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=Intelligence%2FIntegrated%20Forecasting%20System%20,potential%20benefits%20are%20sometimes%20overstated)

. Unlike traditional NWP, ML approaches can directly learn from decades of past weather observations and model outputs. Early successes include systems like GraphCast, a deep learning model that generates 10-day global forecasts in under a minute and **outperforms** the leading physical models on the majority of verification metrics

[6](https://www.science.org/doi/10.1126/science.adi2336#:~:text=GraphCast%2C%20a%20machine%20learning%E2%80%93based%20method,for%20modeling%20complex%20dynamical%20systems)

. This hybrid era – combining physical knowledge with machine learning – represents a new stage in meteorology, complementing the strengths of classical methods with data-driven insights.

## **2. Literature Review**

**ML in Meteorology:** A growing body of research explores using ML algorithms for various weather forecasting tasks. Initial studies applied simple neural networks or decision trees for specific problems like rainfall estimation and wind prediction. For example, neural networks have been used to predict wave heights from time-series data, showing better skill than linear models even with limited data

[7](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=modeling%29,Shahidi)

[8](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=regression%20trees%20%28using%20C5,than%20ANNs%2C%20since%20they%20represent)

. Decision tree methods were also tested in wave forecasting as an interpretable alternative, and they achieved accuracy comparable to neural nets on short datasets

[9](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=note%20that%2C%20as%20with%20ANNs%2C,except%20for%20Bayesian%20networks%2C%20which)

. Over the past decade, the explosion of training data and improved algorithms have led to more powerful models. Convolutional neural networks (CNNs) have been employed for precipitation nowcasting from radar images

[10](https://www.science.org/doi/10.1126/science.adi2336#:~:text=X,30%2C%205617%E2%80%935627%20%282017)

, and recurrent neural networks (RNNs) like LSTM have been used for time-series prediction of temperature and precipitation. Studies in 2021 questioned whether deep learning could rival full NWP systems

[11](https://ijaseit.insightsociety.org/index.php/ijaseit/article/view/18377#:~:text=M,0097)

, and by 2023, ML models like GraphCast demonstrated performance exceeding traditional models in some respects

[12](https://www.science.org/doi/10.1126/science.adi2336#:~:text=GraphCast%2C%20a%20machine%20learning%E2%80%93based%20method,for%20modeling%20complex%20dynamical%20systems)

. This progress has spurred extensive experimentation with different ML architectures in meteorology.

**Hybrid vs. Standalone Models:** Recent literature often compares pure ML models to hybrid approaches that combine multiple techniques. Researchers have investigated hybrid architectures to leverage the advantages of each component model. For instance, CNN–LSTM hybrids (which use CNN for spatial feature extraction and LSTM for temporal modeling) have shown improved accuracy in temperature forecasting compared to CNN or LSTM alone

[13](https://ar5iv.org/pdf/2410.14963#:~:text=combining%20Convolutional%20Neural%20Networks%20,the%20model%27s%20potential%20in%20climate)

. In one study for the Delhi region, a CNN-LSTM model produced a temperature prediction curve that aligned very closely with observations, outperforming single models in both accuracy and stability

[14](https://ar5iv.org/pdf/2410.14963#:~:text=networks%20to%20predict%20historical%20temperature,lays%20the%20groundwork%20for%20future)

. Another example is combining tree-based models with neural networks. A comparative study on energy load forecasting found that integrating an LSTM with XGBoost yielded better results than either model by itself

essay.utwente.nl

. In that study, two hybrid models (LSTM–XGBoost and CNN–XGBoost) were tested against the base XGBoost model; the hybrids **surpassed the standalone model in most categories**

essay.utwente.nl

. Similarly, an ensemble of an RNN (GRU) and XGBoost has been explored for weather sensor data forecasting – the ML models each had strengths in different scenarios, and a combined approach could capture both sequence context and nonlinear relationships

[15](https://ijaseit.insightsociety.org/index.php/ijaseit/article/view/18377#:~:text=Indonesia,model%20was%20better%20since%20it)

. Overall, the literature suggests that no single model type universally dominates; rather, **hybrid models often achieve superior accuracy** by exploiting complementary strengths of different algorithms

essay.utwente.nl

. This trend is observed across domains like rainfall prediction, wind speed forecasting, and power load prediction, indicating broad potential for hybrid ML in weather-related time series forecasting.

## **3. Comparative Analysis of ML Models**

This section compares two popular ML approaches – **Extreme Gradient Boosting (XGBoost)** and **Long Short-Term Memory (LSTM) networks – and examines hybrid combinations** (such as CNN-LSTM or GRU-XGBoost) in terms of interpretability, accuracy, and computational efficiency.

**XGBoost (Extreme Gradient Boosting):** XGBoost is a decision-tree ensemble method that builds many shallow trees in sequence, each correcting errors of the previous ones. It is known for modeling complex nonlinear relationships in structured (tabular) data and includes regularization to prevent overfitting

[16](https://www.mdpi.com/2079-9292/13/14/2719#:~:text=During%20the%20base%20model%20selection,This%20rigorous%20procedure%20ensures%20good)

. XGBoost can naturally handle missing values and provides features like importance scores for each predictor

[17](https://xgboosting.com/xgboost-vs-deep-learning/#:~:text=,scores%2C%20aiding%20in%20model%20interpretation)

, making it relatively interpretable compared to deep neural networks. For example, one can examine the top features contributing to an XGBoost forecast (e.g. humidity or pressure might be ranked highly for predicting rainfall). In terms of accuracy, XGBoost has shown strong performance in many regression tasks. In a weather context, a study on Great Lakes wave forecasting found that a tuned XGBoost model slightly outperformed an LSTM in predicting wave heights, achieving lower error rates

[18](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=and%202017,Lake%20Erie%20yielded%20MAPE%20values)

. Specifically, XGBoost attained a mean absolute percentage error (MAPE) around 16–23% for wave height, compared to 23–31% for the LSTM on the same test set

[19](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=and%202017,Lake%20Erie%20yielded%20MAPE%20values)

. XGBoost was also more precise in wave period prediction than the LSTM in that case. One reason for such success is that tree ensembles can capture nonlinear interactions between weather features (e.g. a combination of temperature, wind, and season) without needing as much data to “learn” the basic relationships. Computationally, XGBoost is efficient to train on CPUs and extremely fast to run for predictions. In the wave height study, the trained XGBoost model was able to generate two years of predictions in **0.03 seconds on a single CPU**

[20](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=although%20it%20did%20not%20improve,03%20s%20on%201%20CPU)

. This speed is a huge advantage for real-time forecasting or for producing large ensembles. The main limitation of XGBoost is that it does not inherently understand temporal order – it treats inputs as vectors, so sequential patterns must be fed in via features (like lagged values or moving averages). This means an expert might need to encode some time-dependence into the input for the model to capture trends or periodicity.

**LSTM (Long Short-Term Memory):** LSTM is a type of recurrent neural network specifically designed to learn from sequential data. An LSTM cell maintains an internal memory state that gets updated over time steps via gated mechanisms

[21](https://d2l.ai/chapter_recurrent-modern/lstm.html#:~:text=Each%20memory%20cell%20is%20equipped,the%20output%20gate)

. There are gates for input, output, and forget; these gates control what information gets added to the cell state, what gets removed, and what gets emitted as output at each time step

[22](https://d2l.ai/chapter_recurrent-modern/lstm.html#:~:text=Each%20memory%20cell%20is%20equipped,the%20output%20gate)

[23](https://d2l.ai/chapter_recurrent-modern/lstm.html#:~:text=The%20data%20feeding%20into%20the,And%20the%20output%20gate)

. By using these gates, LSTMs can retain long-term information and handle the vanishing gradient problem that plagued earlier RNNs. In practical terms, an LSTM can observe a time series of past weather observations (say, the last N hours of temperature, pressure, etc.) and automatically learn which past information is relevant to forecasting the future. LSTMs excel at capturing temporal dependencies – for example, the gradual buildup of a weather pattern or daily cycles – without the user having to manually specify lag features. In terms of accuracy, LSTMs have achieved impressive results in many weather-related tasks. Studies have shown LSTM-based models significantly outperform traditional statistical models in rainfall prediction

[24](https://www.researchgate.net/publication/369179810_An_AI-Enabled_ensemble_method_for_rainfall_forecasting_using_Long-Short_term_memory#:~:text=An%20AI,18%5D%20used)

, and they often beat simpler machine learning methods when long sequences are involved. One research work found that an LSTM trained on 39 years of meteorological data provided more accurate daily precipitation forecasts than multiple linear regression or decision tree models

[25](https://www.researchgate.net/publication/369179810_An_AI-Enabled_ensemble_method_for_rainfall_forecasting_using_Long-Short_term_memory#:~:text=An%20AI,18%5D%20used)

. LSTMs can also handle multivariate inputs (several weather variables over time) and learn complex interactions. However, LSTM networks are largely “black boxes” in terms of interpretability – their learned internal states and weights are not easily understood by humans. This lack of transparency is a disadvantage, especially in meteorology where understanding *why* a model made a certain prediction (e.g., predicting heavy rain) is valuable. Another consideration is computational cost: LSTMs typically require more computational resources (often GPU acceleration) and time to train, especially on long sequences or large datasets. Once trained, though, inference with an LSTM is reasonably fast (the wave model study reported the trained LSTM produced two years of forecasts in 0.24 seconds on one CPU

, which, while slower than XGBoost, is still essentially instantaneous for practical use). Overall, LSTMs bring powerful sequence-learning ability at the expense of interpretability and with a higher training complexity compared to tree ensembles.

**Hybrid Models (CNN-LSTM, GRU-XGBoost, etc.):** Hybrid ML models aim to combine the strengths of different algorithms. A CNN-LSTM, for instance, stacks a Convolutional Neural Network with an LSTM. The CNN can automatically extract spatial features or local patterns (e.g. identifying weather fronts or spatial correlations in gridded data), which are then fed into an LSTM that learns the temporal evolution of those features. This approach has proven effective for problems like climate and weather prediction where both space and time matter. In one study, a hybrid CNN-LSTM model for monthly climate forecasting achieved significantly lower error than either a pure CNN or pure LSTM, reducing the one-month-ahead temperature forecast error substantially

[27](https://www.nature.com/articles/s41598-024-68906-6#:~:text=Monthly%20climate%20prediction%20using%20deep,one%20month%20time%20step%20ahead)

. The authors reported that the hybrid model “significantly reduces the forecasting error” compared to single models

[28](https://www.nature.com/articles/s41598-024-68906-6#:~:text=Monthly%20climate%20prediction%20using%20deep,one%20month%20time%20step%20ahead)

. Another hybrid category pairs tree models with neural networks – for example, using an RNN to capture temporal trends and an XGBoost model to capture any remaining structure or to fine-tune predictions. A 2022 comparative analysis showed that such combined models can yield the best of both worlds: incorporating an LSTM or CNN with XGBoost improved accuracy in most cases tested

essay.utwente.nl

. In that analysis, a CNN-XGBoost hybrid slightly outperformed an LSTM-XGBoost, but both hybrids outperformed a standalone XGBoost baseline

essay.utwente.nl

. The likely reason is that the neural network component can model the sequential dependency that XGBoost alone would miss, while XGBoost can handle nonlinear features and outliers in a way that complements the neural net. Hybrid models, however, tend to be more complex and may require more effort to tune. They also inherit the interpretability challenges of neural nets – for example, a CNN-LSTM’s prediction is as difficult to interpret as a regular LSTM’s, even if the CNN part might identify some known patterns (like “feature maps” for certain weather regimes). One advantage in combinations like XGBoost + LSTM is that the XGBoost component can provide feature importance insights. For instance, if a hybrid model first uses an LSTM to generate features or preliminary forecasts which are then input to XGBoost, the XGBoost’s analysis might tell us which signals from the LSTM (or which original input features) were most influential. In summary, hybrid models generally improve **accuracy** – often achieving lower RMSE or higher skill than single-model counterparts – but they do so at a cost of higher **complexity**. Table 1 provides a high-level comparison of these ML approaches.

**Table 1. Comparison of ML Model Approaches in Weather Forecasting**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Interpretability** | **Notable Strengths** | **Notable Limitations** | **Example Performance** |
| **XGBoost** | Moderate – can inspect feature importance; decision rules are somewhat interpretable  . | Excels at capturing nonlinear relationships in tabular data; fast training and inference; handles missing data and outliers well    . | Does not inherently model time dependencies (needs lag features); may require many trees for very complex patterns. | MAPE ~15–23% for wave height vs 23–31% for LSTM (Lake Erie study)  . In a load forecast, XGBoost outperformed another tree model (CatBoost) on accuracy  essay.utwente.nl  . |
| **LSTM** | Low – acts as a “black box” (internal states not easily understood)  [xgboosting.com](https://xgboosting.com/xgboost-vs-deep-learning/#:~:text=,boxes%E2%80%9D%20due%20to%20their%20complexity)  . | Learns long-term temporal dependencies automatically; suitable for sequence data (captures trends, seasonality); can handle multivariate time series. | Training can be slow and data-hungry; prone to overfitting without care; requires more computational power (e.g. GPUs). | Significantly outperformed linear regression for daily rainfall prediction  [researchgate.net](https://www.researchgate.net/publication/369179810_An_AI-Enabled_ensemble_method_for_rainfall_forecasting_using_Long-Short_term_memory#:~:text=An%20AI,18%5D%20used)  . Slightly less accurate than XGBoost for Lake Erie wave heights  [repository.library.noaa.gov](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=and%202017,Lake%20Erie%20yielded%20MAPE%20values)  , but often better for longer-term or more complex sequence tasks. |
| **CNN–LSTM** (hybrid) | Low – highly complex deep model, difficult to interpret individual features. | Combines spatial pattern recognition (CNN) with temporal modeling (LSTM); effective for spatio-temporal data (e.g. maps evolving in time). | High complexity; requires large data and careful tuning; computationally intensive to train. | More accurate than CNN or LSTM alone for temperature forecasting  [ar5iv.org](https://ar5iv.org/pdf/2410.14963#:~:text=combining%20Convolutional%20Neural%20Networks%20,the%20model%27s%20potential%20in%20climate)  (hybrid model’s predictions closely matched actual trends). Used for climate forecasts to reduce error by leveraging both space and time features  [nature.com](https://www.nature.com/articles/s41598-024-68906-6#:~:text=Monthly%20climate%20prediction%20using%20deep,one%20month%20time%20step%20ahead)  . |
| **GRU–XGBoost** (hybrid) | Low–Moderate – the GRU (a type of RNN) is black-box, but XGBoost output can be interpreted to some extent. | Captures sequential patterns via GRU and fine-tunes with XGBoost; can improve peak prediction and overall accuracy by blending sequence learning with feature-based learning. | Model integration is complex; two-stage training might be needed; still not fully transparent; potential for higher computational cost. | Hybrid models (GRU/CNN + XGBoost) **surpassed** standalone models in most categories of a load forecasting study  essay.utwente.nl  . In one weather study, XGBoost beat GRU for short-term prediction, but GRU had an edge in longer-range context  [ijaseit.insightsociety.org](https://ijaseit.insightsociety.org/index.php/ijaseit/article/view/18377#:~:text=Indonesia,model%20was%20better%20since%20it)  , suggesting the hybrid could capture both. |

**Interpretability vs Accuracy:** It is evident that there is often a trade-off between model interpretability and pure accuracy. Simpler models or tree-based models (like XGBoost) allow more interpretation (e.g., rules or feature rankings), whereas LSTM and deep hybrids generally offer higher accuracy on complex tasks but operate as opaque systems. For critical applications, this trade-off must be managed carefully – sometimes a slightly less accurate but interpretable model might be preferred for trust and insight.

## **4. Comparison of ML and Physical Forecasting Models**

With the success of machine learning in many forecasting tasks, an important question is how ML-based approaches compare to traditional physical models in weather forecasting. There are fundamental differences between the two:

**Basis of Predictions:** Physical NWP models are grounded in well-understood science – they numerically solve equations for atmospheric dynamics and thermodynamics. This gives them an advantage in adhering to known physical laws (e.g., conservation of energy, mass continuity) and in handling unprecedented scenarios by pure physics reasoning. ML models, by contrast, are purely data-driven: they learn statistical patterns from historical data. They do not “know” the laws of physics explicitly, so they might propose solutions that are meteorologically inconsistent (like violating a physical constraint) unless constrained or guided by some prior knowledge. However, because ML models learn directly from data, they can correct systematic biases present in NWP outputs and can make use of the full history of observations to inform predictions – something traditional models don’t directly do

[science.org](https://www.science.org/doi/10.1126/science.adi2336#:~:text=Global%20medium,better%20severe%20event%20prediction%2C%20including)

. For example, an NWP model might consistently underestimate afternoon temperatures in a region due to terrain resolution issues; an ML model could learn this bias from past forecasts vs observations and adjust accordingly.

**Accuracy and Skill:** In recent comparisons, advanced ML models have begun to rival or exceed the accuracy of operational NWP in certain contexts. A striking example is **GraphCast** (by Google DeepMind), which is a learned weather simulator. GraphCast was shown to *significantly outperform* the **most accurate operational deterministic** NWP systems on 90% of key weather metrics in a 10-day global forecast setting

[science.org](https://www.science.org/doi/10.1126/science.adi2336#:~:text=resolution%20globally%20in%20under%201,atmospheric%20rivers%2C%20and%20extreme%20temperatures)

. It was able to better predict the tracks of tropical cyclones, atmospheric rivers, and extreme temperature events than the ECMWF high-resolution model in that study

[science.org](https://www.science.org/doi/10.1126/science.adi2336#:~:text=GraphCast%2C%20a%20machine%20learning%E2%80%93based%20method,for%20modeling%20complex%20dynamical%20systems)

. Moreover, GraphCast did this at a tiny fraction of the runtime (minutes instead of hours). This demonstrates that, given enough data and training, ML models can encapsulate a lot of the physics implicitly and achieve high skill. On the other hand, traditional models still excel in scenarios where physical consistency and extrapolation are crucial. For instance, when predicting a truly novel event outside the range of historical examples (say, an unusual combination of meteorological factors), a physics-based model can still churn out a plausible forecast based on equations, whereas an ML model might be uncertain or give an unreliable output if it hasn’t seen a similar pattern in training.

In practice, rather than choosing one over the other, a hybrid or complementary use is emerging. Many weather services use **post-processing** ML models to improve NWP forecasts. For example, MOS (Model Output Statistics) is a longstanding approach where statistical models (now often ML) correct biases in NWP output. ML models can also downscale coarse model output to local forecasts or infer variables that the NWP might not predict directly (like solar irradiance for a solar farm, derived from NWP-predicted cloud cover using ML). Meanwhile, the NWP provides physically reliable baseline forecasts and can simulate extreme scenarios for which data is sparse.

**Strengths and Weaknesses:** Traditional models have the strength of **theoretical soundness** and **interpretability for meteorologists**. Forecasters can examine NWP outputs (like pressure maps, wind fields) and understand the meteorological reasoning (e.g., a cold front causing rain). These models also provide a full spatial and vertical picture of the atmosphere, which purely data-driven models might not inherently do unless they are designed to. Their weaknesses include high computational cost and spin-up time – running a global high-resolution model might take hours on a supercomputer, and small changes in initial conditions can lead to divergent outcomes (the “butterfly effect”). ML models, conversely, run extremely fast once trained and can be updated almost instantly with new data. The Lake Erie wave study illustrated this well: the physical wave model (WAVEWATCH III) took 24 hours on 60 CPUs to produce a 2-year hindcast, while the trained ML models produced the same output in seconds on a single CPU

[repository.library.noaa.gov](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=although%20it%20did%20not%20improve,03%20s%20on%201%20CPU)

. Speed allows for rapid updates and large ensembles of predictions to quantify uncertainty without massive computing costs. However, a key weakness of many current deterministic ML forecasts is the lack of **uncertainty quantification** – physical modeling centers produce ensemble forecasts (dozens of model runs with slightly varied conditions) to estimate forecast confidence, whereas a single ML model typically gives one deterministic answer. Efforts are underway to create ML ensemble methods or probabilistic ML forecasts to address this.

Another aspect is **generalization**. NWP models, because they rely on fundamental equations, can in principle handle climate shifts (they can be run under different atmospheric compositions, etc., and still solve the same physics). ML models trained on past climate data might struggle if climate change pushes weather patterns beyond what they’ve seen historically. This is why some experts approach the recent hype with caution

[dexterenergy.ai](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=its%20IFS%20weather%20forecast%20model,potential%20benefits%20are%20sometimes%20overstated)

– while deterministic AI models are a huge advancement, over-reliance on them without understanding their failure modes could be risky.

**Real-world Applications:** We are seeing a convergence in real-world systems. For instance, ECMWF’s new AIFS (AI/Integrated Forecasting System) mentioned earlier is essentially using ML to emulate its physical model

[dexterenergy.ai](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=After%202023%20turned%20out%20to,potential%20benefits%20are%20sometimes%20overstated)

– a form of *model compression*. The idea is to run an ML model that mimics the full NWP, getting almost the same result much faster. Other applications include using deep learning to extrapolate radar images for very short-term precipitation forecasts (nowcasting), where ML methods (like CNNs/LSTMs) significantly outperform NWP for 0-2 hour predictions of storm intensity

[science.org](https://www.science.org/doi/10.1126/science.adi2336#:~:text=X,30%2C%205617%E2%80%935627%20%282017)

. In contrast, for medium-range (5-10 day) outlooks, hybrids of ML and NWP might be best: ML provides speed and possibly improved accuracy on average, but NWP contributes reliable structure and the ability to simulate edge cases.

In summary, ML-based approaches are no longer just experimental – they are becoming competitive with physical models in accuracy and clearly superior in computational efficiency in some cases. Traditional NWP remains indispensable for its physical fidelity and well-tested reliability, especially for forecasting complex 3D atmospheric structures and handling novel situations. The likely future of weather forecasting is a combination of the two: using ML to **enhance and accelerate** physical models (for example, AI-assisted ensembles or ML-corrected forecasts

[dexterenergy.ai](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=Additionally%2C%20these%20models%20seem%20to,effective%20to%20run)

), thereby getting the best accuracy possible while managing computational cost and maintaining confidence in the physical realism of the forecast.

## **5. Hypotheses & Research Objectives**

**Hypotheses:**

1. *Hybrid machine learning models (such as an XGBoost+LSTM combination) will achieve higher forecast accuracy than standalone models.* This hypothesis is based on the idea that hybrid models can capture complementary aspects of the data. The tree-based component (XGBoost) can effectively model nonlinear relationships and handle irregular variations, while the LSTM captures temporal dependencies and sequence patterns

[mdpi.com](https://www.mdpi.com/2079-9292/13/14/2719#:~:text=performance,Lasso%20regression)

. By combining them, we expect the hybrid to outperform either model alone, as observed in prior studies where hybrid models surpassed single learners

essay.utwente.nl

.

1. *Hybrid models can provide a balance of interpretability and performance, potentially making them more practical for operational use than pure deep learning models.* Here we hypothesize that including an interpretable element (like XGBoost) in the model can help retain some transparency (e.g., feature importance on the XGBoost input) without sacrificing much accuracy. While the overall hybrid is still complex, any insight from one part of the model could be valuable. We also expect that the hybrid will still be faster to run than physical models, and possibly as fast as simpler ML models at inference time, making it suitable for real-time forecasting.
2. *Machine learning-based forecasting, whether hybrid or standalone, can match the performance of traditional numerical models for certain weather parameters.* This hypothesis will be explored by comparing forecast error statistics. For example, prior evidence showed an ML model equaling or beating a physics-based wave model in accuracy for wave height prediction

[repository.library.noaa.gov](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=and%202017,yielded%20lower%20scatter%20scores%20than)

. We aim to test if that holds for other variables (like temperature or rainfall), and under what conditions (data size, forecast horizon) ML models excel or fall short compared to NWP outputs.

**Research Objectives:**

* **Objective 1:** Evaluate and compare the performance of XGBoost, LSTM, and a hybrid XGBoost+LSTM model for a specific weather forecasting task (e.g., predicting daily temperature or precipitation). Performance will be measured using standard metrics like RMSE, MAE, and R² (defined in Section 8) to quantitatively assess accuracy. This addresses the question of whether the hybrid model truly offers an improvement over the standalone models.
* **Objective 2:** Analyze the interpretability and computational efficiency of each modeling approach. This involves examining model outputs (e.g., feature importance from XGBoost, learned patterns from LSTM) and measuring training/inference time. The goal is to understand practical trade-offs – for instance, is the hybrid model computationally feasible for operational use, and does it offer any interpretability advantages through the XGBoost component?
* **Objective 3:** Compare the ML models’ forecasts with a baseline physical model or official forecast. By doing so, we seek to contextualize the ML models’ performance in a real-world scenario. For example, if data is available from a local NWP or a persistence forecast, we will compare our models against that baseline to see if ML provides added value. This helps determine if (and by how much) machine learning can improve weather prediction accuracy in practice.
* **Objective 4:** Identify the challenges and limitations encountered by the ML models in forecasting weather, and suggest improvements. This includes examining cases where the models err significantly (e.g., sudden weather regime changes or extreme events) to diagnose whether the issue is due to data deficiencies, model limitations, or inherent unpredictability. The aim is to derive insights that could guide future hybrid model development or data augmentation for better forecasts.

By addressing these objectives, the study will provide a comprehensive comparison of hybrid and standalone ML models in weather forecasting, shedding light on their practicality and effectiveness relative to each other and to traditional methods.

## **6. Weather Forecasting Methods**

Weather forecasting methods can be broadly categorized into two groups: traditional (physics-based and statistical) approaches and modern machine learning approaches. Below is an overview of key methods in each category:

* **Numerical Weather Prediction (NWP):** This is the traditional approach, where mathematical models of the atmosphere are run on computers to simulate future weather. NWP models like ECMWF’s IFS, NOAA’s GFS, the UK Met Office model, etc., solve equations for fluid flow, thermodynamics, radiative transfer, and other physical processes

[dexterenergy.ai](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=Let%E2%80%99s%20start%20with%20the%20%E2%80%98traditional%E2%80%99,surface%20to%20the%20upper%20atmosphere)

[dexterenergy.ai](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=Some%20of%20the%20most%20well,global%20deterministic%20NWP%20models%20are)

. They divide the atmosphere into a 3D grid and compute how weather variables evolve at each grid point. NWP requires assimilation of current observations to initialize the model, and then time-stepping the equations forward. These models produce detailed forecasts and are the foundation for most weather agencies. Variants include global models (covering the whole Earth at moderate resolution) and regional high-resolution models that zoom into specific areas for finer detail

[dexterenergy.ai](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=As%20their%20name%20indicates%2C%20high,have%20multiple%20levels%20of%20nesting)

.

* **Ensemble Forecasting:** Rather than a single deterministic forecast, ensemble forecasting runs multiple simulations with slight tweaks (e.g., varying initial conditions within their uncertainty range, or using different model physics options). The idea is to capture the range of possible outcomes, addressing the inherent uncertainty in weather systems. For example, ECMWF’s Ensemble Prediction System (ENS) runs 50 perturbed forecasts alongside its main run

[dexterenergy.ai](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=Image)

. The result is a collection of forecasts from which probabilistic predictions can be made (e.g., a 70% chance of rain, or a spread of possible hurricane tracks). Ensemble methods have become a standard part of weather forecasting to convey confidence levels.

* **Statistical and Analog Methods:** Before modern ML, forecasters used various statistical techniques to improve or interpret model forecasts. **Model Output Statistics (MOS)** is one such method where historical observations are regressed against NWP forecast variables to develop correction equations. Simpler approaches like linear regression, persistence (assuming conditions will remain the same as current in the short term), or climatology (using long-term average conditions as a “forecast”) are also baseline methods. **ARIMA (Auto-Regressive Integrated Moving Average)** and related time-series models have been applied to meteorological data for short-term forecasting of things like air quality or river flow, and even to adjust load forecasts using weather inputs

[mdpi.com](https://www.mdpi.com/2079-9292/13/14/2719#:~:text=A%20statistical%20model%20is%20based,been%20proposed%20to%20further%20enhance)

[mdpi.com](https://www.mdpi.com/2079-9292/13/14/2719#:~:text=mathematical%20functions%20and%20statistical%20principles,141)

. Analogs (finding past situations similar to the current one and seeing what happened) have also been used historically for forecasting specific patterns, though these methods are largely supplanted by more advanced techniques now.

* **Machine Learning-Based Techniques:** This category includes a variety of data-driven models:
  + **Decision Trees and Ensemble Trees:** Methods like Random Forests and Gradient Boosting (XGBoost, LightGBM, etc.) fall here. They work well with structured data, including meteorological features (temperature, humidity, pressure, etc.) at various times or locations as inputs. They can handle nonlinear effects (for example, a sudden jump in rainfall once humidity crosses a threshold combined with certain temperature) and are relatively robust. These have been used for tasks like forecasting solar radiation from weather inputs or predicting severe weather occurrences. Tree-based models do not need the user to specify interactions – they discover them – and they provide some interpretability (feature importance, decision rules) which is useful

[repository.library.noaa.gov](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=note%20that%2C%20as%20with%20ANNs%2C,except%20for%20Bayesian%20networks%2C%20which)

.

* + **Artificial Neural Networks (ANNs):** These range from simple multi-layer perceptrons (fully-connected networks) to more complex architectures. In meteorology, even early studies found that ANNs could slightly outperform traditional regression for things like wind and wave prediction

[repository.library.noaa.gov](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=regression%20trees%20%28using%20C5,than%20ANNs%2C%20since%20they%20represent)

. They take numerical inputs (which could be current or recent past observations from various sensors) and learn to map to the target output (like tomorrow’s temperature). ANNs can model nonlinearities but treat data as independent samples unless combined with recurrence or convolution to capture structure.

* + **Convolutional Neural Networks (CNNs):** CNNs are adept at handling spatial data, like weather maps or satellite images. A CNN can be used to predict the next radar image frame given the last few frames (a technique for short-term rainstorm forecasting). CNNs in weather have been used for precipitation nowcasting

[science.org](https://www.science.org/doi/10.1126/science.adi2336#:~:text=X,30%2C%205617%E2%80%935627%20%282017)

, climate pattern recognition, and even detecting features like tropical cyclones or fronts in model outputs. They excel at recognizing localized patterns (for instance, the shape of rainfall clusters).

* + **Recurrent Neural Networks (RNNs):** These include LSTM and GRU networks, which are specialized for sequential data. In weather forecasting, RNNs can use time-series data (like the past 10 days of weather observations or predictions) to forecast the next day(s). They implicitly account for temporal correlations. LSTMs have been widely applied for things like forecasting temperature, precipitation, wind speed, and atmospheric indices because they can model seasonality and lagged effects automatically. For example, an LSTM can learn that an extended period of dry days increases the chance that the next day’s humidity will be low, etc., without being explicitly told the rule.
  + **Hybrid Models:** As discussed in Section 3, hybrids combine elements of the above (e.g., CNN-LSTM, or an ANN to post-process NWP output). Another form of hybrid is using ML to **augment physical models** – for example, using a neural network to predict sub-grid processes in an NWP model (a technique known as “neural parameterization”), or to emulate an entire physics model (like AIFS does for IFS

[dexterenergy.ai](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=After%202023%20turned%20out%20to,potential%20benefits%20are%20sometimes%20overstated)

). Such approaches blend physical modeling with learning from data, aiming to get the accuracy of the former with the speed of the latter.

* + **Other ML techniques:** Support Vector Machines (SVMs) and Gaussian Process Regression have also been explored for certain meteorological problems (e.g., SVMs for classifying weather regimes or rain vs no-rain). Additionally, clustering algorithms (k-means, etc.) might be used in pre-processing to categorize similar weather patterns, which can then feed into predictive models.

Each method has its niche. In practice, operational forecasting uses an ensemble of methods: NWP provides a primary forecast, ensembles give uncertainty bounds, statistical models correct biases, and ML models are increasingly used to refine outputs or speed up certain predictions. The choice of method often depends on the forecast horizon (for very short term, ML can be advantageous; for long term, physical models plus ML post-processing might work best) and the variable of interest (for example, predicting a binary event like thunderstorm occurrence might use a different approach than predicting a continuous variable like temperature).

## **7. Problem Statement & Critical Analysis**

**Problem Importance:** Improving weather forecasting accuracy has significant societal and economic benefits. Weather affects agriculture, transportation, energy, public safety, and virtually all outdoor activities. Even modest improvements in forecast skill can translate into better preparedness and optimization. For instance, in the renewable energy sector, more accurate wind and solar forecasts make power generation more predictable, helping balance supply and demand on the grid

[dexterenergy.ai](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=Accurate%20weather%20forecasts%20are%20fundamental,for%20power%20producers%20or%20traders)

. This reduces costs and the risk of power outages. In terms of safety, better storm forecasts save lives by enabling timely evacuations or advisories. Accurate forecasts of heavy rainfall can mitigate flood damage by prompting early water management actions. Thus, there is a continuous drive to reduce forecast errors (like lowering RMSE for temperature or precipitation forecasts) and to extend reliable forecast lead times. Furthermore, as extreme weather events become more frequent with climate change, improving forecast accuracy (especially for extremes) is critical for disaster resilience. This study addresses the problem of forecast accuracy by examining whether modern ML techniques, especially hybrid models, can push the boundaries of predictive performance beyond what single models or traditional methods achieve.

**Challenges in Improving Accuracy:** Despite advancements, several challenges persist:

* **Data Availability and Quality:** High-quality historical data is the fuel for ML models. While there are extensive records (e.g., global reanalysis datasets spanning decades), not all weather variables are equally well observed. Some regions (like over oceans or in developing countries) have sparse observations, which can limit model training. Additionally, different data sources may have biases or inhomogeneities (due to changing instruments, etc.). ML models might latch onto spurious correlations if the training data isn’t carefully curated. For example, if an observing station moved and caused a shift in recorded temperature, a naive model might interpret that as a genuine climate signal. Another data challenge is the sheer volume – global high-resolution data can be terabytes in size, making training difficult. However, techniques and infrastructure for big data are rapidly improving, allowing training on large weather datasets like ERA5 reanalysis which spans 40+ years

[dexterenergy.ai](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=In%20essence%2C%20AI%20models%20emulate,v5%2C%20commonly%20known%20as%20ERA5)

.

* **Chaotic Nature of Weather:** The atmosphere is a chaotic system; tiny changes can lead to diverging outcomes (the butterfly effect). This inherently limits predictability, no matter the method. Even a perfect model cannot forecast beyond a certain horizon for small-scale features. ML models face the same fundamental limit – they may struggle with inherently unpredictable events or sudden transitions. If not designed to handle uncertainty, an ML model might be overconfident in such situations. That’s why ensemble methods (whether physical or ML) are important for gauging uncertainty. A critical analysis must acknowledge that improved accuracy on average doesn’t eliminate the unpredictability of certain events.
* **Model Generalization and Overfitting:** ML models, especially highly flexible ones, risk overfitting the historical data. They might memorize the training examples rather than learning general patterns. In weather forecasting, overfitting could mean a model works very well for past years but fails to generalize to future unseen scenarios. Careful validation is needed, and techniques like cross-validation over different periods, regularization (for XGBoost this is built-in, for neural nets techniques like dropout can help), and early stopping are used to prevent this. Additionally, non-stationarity (climate trends) means the underlying data distribution is slowly changing; models need to be updated periodically to account for current climate conditions.
* **Interpretability and Trust:** As forecasts improve with ML, a parallel concern is the ability to interpret and *trust* these predictions. Weather forecasters and stakeholders often need to know the reasoning behind a forecast, especially if it’s an outlier or contradicts other guidance. Traditional models produce outputs that can be traced to meteorological features (e.g., a forecast map might clearly show a cold front leading to precipitation – something a human can understand). ML models might output a number with no explanation. The lack of interpretability can be a barrier to adoption: meteorologists may be hesitant to use a black-box forecast for critical decisions. Recent work on interpretability is trying to address this by applying methods like SHAP values or saliency maps to weather ML models

[sciencedirect.com](https://www.sciencedirect.com/science/article/abs/pii/S1352231024004722#:~:text=prediction%20www,CAM)

[agupubs.onlinelibrary.wiley.com](https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2023GL107377#:~:text=,1)

. Still, the complexity of hybrid models adds layers of difficulty in explanation. This study critically evaluates if the accuracy gains from hybrids justify the loss in interpretability, and if there are ways to extract meaningful insights from them (for example, examining which features the XGBoost part is using the most, as a proxy for what drives the forecast).

* **Computational Cost:** While ML models can be very fast at runtime, training them can be computationally expensive, especially deep learning models. Training may require specialized hardware (GPUs/TPUs) and can take hours to days for large datasets. There is also the energy cost aspect – big ML models consume a lot of power to train, leading to concerns about carbon footprint (this is also a concern for large NWP ensembles, of course). In operational settings, there is a need to retrain models periodically to incorporate new data, which adds to the cost. However, after training, ML models like GraphCast are extremely efficient in production

[science.org](https://www.science.org/doi/10.1126/science.adi2336#:~:text=GraphCast%2C%20a%20machine%20learning%E2%80%93based%20method,for%20modeling%20complex%20dynamical%20systems)

. Physical models, conversely, have a fixed cost for each run. There’s an interesting trade-off: NWP requires heavy computation every time you run a forecast, whereas an ML model has a one-time training cost and then relatively cheap forecasts. For national weather centers, integrating ML could mean rethinking their computational resource allocation (potentially less CPU for NWP, more GPU for ML training).

* **Integration with Existing Systems:** Another practical challenge is integrating ML forecasts with existing forecasting workflows. Meteorologists are used to certain data formats, lead times, and types of guidance. If an ML model predicts something very different from the trusted NWP, how should that be used? Blending ML outputs with traditional forecasts in a coherent way is a challenge. One approach is to use ML as just another member in an ensemble or as a bias correction tool. Our study’s problem analysis includes considering these operational hurdles: it’s not enough for a model to be accurate in tests; it must also be deployable and complement other tools.

In summary, the push for better forecasting accuracy via hybrid ML models is motivated by high-impact benefits, but it comes with challenges in data, model behavior, and practical deployment. This research is not just about building a more accurate model; it critically examines these challenges. By understanding where the hybrid model succeeds and where it struggles (for example, does it fail on rare extreme events? does it mislead when outside its training domain?), we can identify future improvements. For instance, the study might find that adding physical constraints or using an ensemble of ML models helps address some weaknesses. The ultimate goal is to contribute to more reliable and accurate weather forecasts that can be confidently used in decision-making.

## **8. Evaluation Metrics**

To quantitatively assess forecasting models, we use several standard regression evaluation metrics: **Root Mean Square Error (RMSE)**, **Mean Absolute Error (MAE)**, and **R² (R-squared, the coefficient of determination)**. Each metric highlights different aspects of performance and is important for understanding model accuracy in a weather prediction context.

* **Mean Absolute Error (MAE):** MAE measures the average magnitude of the errors without considering their direction

[resources.eumetrain.org](https://resources.eumetrain.org/data/4/451/english/msg/ver_cont_var/uos3/uos3_ko1.htm#:~:text=The%20MAE%20measures%20the%20average,weighted%20equally%20in%20the%20average)

. In other words, it’s the mean of the absolute differences between predicted values and observed values. If a temperature forecast has an MAE of 2°C, it means that on average, the forecast is 2°C off from the actual temperature (regardless of being higher or lower). MAE is a linear score – each error influences the metric in direct proportion to its absolute size

[resources.eumetrain.org](https://resources.eumetrain.org/data/4/451/english/msg/ver_cont_var/uos3/uos3_ko1.htm#:~:text=the%20differences%20between%20forecast%20and,weighted%20equally%20in%20the%20average)

. This means all errors are weighted equally; a 5° error is exactly five times worse than a 1° error in the eyes of MAE. MAE is easy to interpret (being in the same units as the variable), which is a reason it’s popular. In weather forecasting, MAE tells you the typical error to expect. For instance, an MAE of 1.5 mm in a daily rainfall forecast suggests that, on average, the forecasted rainfall amount is off by 1.5 mm from reality – a relatively small error if typical rainfall is, say, 10 mm, but a larger concern if typical rainfall is 2 mm. MAE is useful for a general sense of accuracy, but it does not penalize occasional large errors any more than frequent small errors, which brings us to RMSE.

* **Root Mean Square Error (RMSE):** RMSE is defined as the square root of the average of squared errors between forecasts and observations

[resources.eumetrain.org](https://resources.eumetrain.org/data/4/451/english/msg/ver_cont_var/uos3/uos3_ko1.htm#:~:text=The%20RMSE%20is%20a%20quadratic,large%20errors%20are%20particularly%20undesirable)

. Because errors are squared before averaging, RMSE gives more weight to large errors

[resources.eumetrain.org](https://resources.eumetrain.org/data/4/451/english/msg/ver_cont_var/uos3/uos3_ko1.htm#:~:text=square%20root%20of%20the%20average,large%20errors%20are%20particularly%20undesirable)

. For example, consider two forecast methods over four days: Method A has errors of [0, 0, 0, 8] mm (three perfect days and one day it rained 8 mm and was missed), Method B has errors of [2, 2, 2, 2] mm. Both have an MAE of 2 mm. But Method A’s RMSE would be 4 mm, while Method B’s RMSE is about 2 mm. This reflects that Method A’s one large miss is particularly undesirable. In weather forecasting, RMSE is often the primary metric because we care a lot about avoiding big mistakes (missing a storm, for instance). A lower RMSE indicates a model has fewer large errors and overall is more consistently close to the truth

[medium.com](https://medium.com/@vinay1996/the-top-11-timeseries-forecasting-metrics-you-need-to-know-mape-smape-me-mae-rmse-mse-5023fd26aa88#:~:text=MAPE%20medium,the%20forecasted%20values%20are)

. One thing to note: RMSE will always be ≥ MAE, and the difference between them grows when there are outlier errors

[resources.eumetrain.org](https://resources.eumetrain.org/data/4/451/english/msg/ver_cont_var/uos3/uos3_ko1.htm#:~:text=The%20MAE%20and%20the%20RMSE,are%20of%20the%20same%20magnitude)

. Meteorologists often use RMSE for continuous variables like temperature, wind speed, or pressure. For example, the RMSE of a 24-hour temperature forecast might be ~2°C for a good model – meaning that overall, the model’s error has a standard deviation of 2°C. Lowering RMSE by even 0.1°C in such a case could be statistically significant over many forecasts and is a win for model improvement.

* **R² (Coefficient of Determination):** R² represents the proportion of variance in the observed data that is explained by the model’s predictions

[scribbr.com](https://www.scribbr.com/statistics/coefficient-of-determination/#:~:text=You%20can%20interpret%20the%20coefficient,predicted%20by%20the%20statistical%20model)

. It ranges from 0 to 1 (or 0% to 100%). An R² of 0.9 (90%) means that 90% of the variability in, say, daily high temperature is explained by the model, and only 10% is left in the residual (unexplained). In formula terms, R² = 1 – (SS\_residual / SS\_total), where SS\_total is the total variance of the observations from their mean, and SS\_residual is the variance of the forecast errors

[scribbr.com](https://www.scribbr.com/statistics/coefficient-of-determination/#:~:text=Image%3A%20%5Cbegin%7Bequation%2A%7DR%5E2%3D1)

[scribbr.com](https://www.scribbr.com/statistics/coefficient-of-determination/#:~:text=You%20can%20interpret%20the%20coefficient,predicted%20by%20the%20statistical%20model)

. R² is a convenient single measure of goodness-of-fit. In weather terms, high R² usually correlates with low MSE (mean square error), but it is more interpretable in a variance-explained sense. However, R² can be less intuitive for meteorological time series because even a perfect forecast of a somewhat chaotic variable might not get R² = 1 due to inherent unpredictability or small-scale noise. Also, R² can be misleading if used with non-linear models or if the mean of the data is tricky (for example, if there’s a strong seasonal cycle, a model that at least predicts the seasons right will get a high R² even if day-to-day variability is off). Despite those caveats, R² is useful for comparing models: a higher R² means a model’s predictions line up more closely with observed variations. For instance, if Model X has R²=0.8 for daily rainfall and Model Y has R²=0.6, it suggests Model X captures more of the day-to-day fluctuation in rainfall amounts (perhaps it nails the distinction between wet and dry days better).

**Why These Metrics Matter:** In weather forecasting, communicating and evaluating error is essential. RMSE and MAE directly indicate how wrong a forecast tends to be, which has practical implications (a 2°C temperature MAE might be fine for planning your outfit, but a 2°C RMSE in predicting freezing vs non-freezing can determine road treatment for ice). RMSE’s emphasis on large errors aligns with the goal to avoid big misses in forecasts (for example, failing to predict a severe storm is much worse than a slight drizzle intensity error). MAE gives a more robust sense of typical error that is less sensitive to outliers; forecasters might look at MAE to understand everyday performance. R² provides an overall skill score – useful in research to compare how well different models capture variability. It is common in academic studies to report R² especially when dealing with new modeling techniques, to show how much variance is explained by the new method.

For our comparative study, we will use these metrics to evaluate each model (XGBoost, LSTM, hybrid). For example, we might find that the hybrid model has an RMSE 10% lower than the standalone LSTM – a notable improvement. We will also consider metric differences in different situations (does one model excel in MAE but not in RMSE? That could indicate it usually does well but occasionally has big errors). Moreover, some metrics may be more relevant depending on the variable: e.g., R² is often reported for temperature (where variance is high between summer and winter), whereas for something like wind direction (a circular variable) we might rely on a different error metric altogether. In this study, focusing on RMSE, MAE, and R² gives a comprehensive picture: MAE and RMSE for absolute error magnitude, and R² for how well the pattern of ups and downs is captured.

## **9. Understanding LSTM and XGBoost**

To better appreciate the capabilities of the models we are studying, it’s important to understand **how LSTM and XGBoost work** and what their strengths and limitations are, specifically in the context of weather forecasting.

**LSTM (Long Short-Term Memory) Networks:** LSTM is an advanced type of recurrent neural network designed to handle long-term dependencies. A standard RNN would simply take the previous hidden state and current input to produce a new state, which often leads to issues like vanishing gradients (where influence of older inputs fades away). LSTM addresses this with a special cell structure featuring gates. Each LSTM cell has an internal memory **state** that runs through time steps, and three primary gates that control the flow of information: an **input gate**, a **forget gate**, and an **output gate**

[d2l.ai](https://d2l.ai/chapter_recurrent-modern/lstm.html#:~:text=Each%20memory%20cell%20is%20equipped,the%20output%20gate)

[d2l.ai](https://d2l.ai/chapter_recurrent-modern/lstm.html#:~:text=The%20data%20feeding%20into%20the,And%20the%20output%20gate)

.

* The **input gate** decides how much of the new input to write into the cell’s memory.
* The **forget gate** determines how much of the past information to erase (or keep). If the forget gate output is 1, the old information is kept fully; if 0, it’s completely dropped, and values between 0-1 allow partial retention

[d2l.ai](https://d2l.ai/chapter_recurrent-modern/lstm.html#:~:text=Each%20memory%20cell%20is%20equipped,the%20output%20gate)

.

* The **output gate** controls how much of the internal state is revealed to the next layer or as output.

Through this mechanism, LSTMs can keep important information in memory for many time steps and ignore irrelevant inputs. For example, in a weather sequence, an LSTM could learn to “remember” a persistent high-pressure system’s presence over many days (since it might be crucial for temperature forecasts), but “forget” short-term noise like a temporary sensor glitch or a minor fluctuation that doesn’t affect the overall trend. Essentially, LSTM learns what to remember and what to forget.

In practice, when we feed an LSTM a time series of, say, daily temperatures and humidity, it updates its cell state at each time step considering those inputs and its prior state. By training on many sequences, it learns patterns like diurnal cycles, seasonal cycles, or the typical duration of certain weather events (like how long a heatwave usually lasts, for instance). LSTMs are well-suited for meteorological data because weather has memory – today’s weather is usually strongly related to yesterday’s (except in very volatile situations). LSTMs automatically exploit this continuity.

**Strengths of LSTM:** The primary strength is capturing **long-term temporal dynamics**. It can handle sequences with dozens or hundreds of steps, making it ideal for incorporating information from days or weeks of past observations into a forecast. In weather, this means it could learn, for example, that after a week of increasing soil moisture, the probability of heavy rainfall might change (due to land-atmosphere feedbacks), or that certain oscillations (like the Madden-Julian Oscillation in tropics) have multi-week periods that influence local weather. LSTMs also handle multivariate inputs; they can ingest multiple features over time (temperature, humidity, pressure, etc.) and find interdependencies (maybe it learns that a drop in pressure combined with high humidity sustained over time leads to rainfall).

**Limitations of LSTM:** They require a lot of data to train effectively, especially if the sequences are long, to cover all the patterns. Training is computationally intensive – training an LSTM on decades of hourly weather data is a big task. There’s also a risk of overfitting to particular sequences if not enough variability is present in training. Another limitation is that LSTMs, by themselves, don’t provide insight into *which* aspects of the sequence were most important – they’re a black box mapping from input sequence to output. One might use techniques to peek inside (like looking at the gates’ values or using attribution methods), but it’s not straightforward. In forecasting, if an LSTM predicts an extreme event, it might be hard to explain whether it was a specific combination of factors or some long-term buildup that led it to that conclusion. From an operational point of view, LSTMs also might not inherently respect physical constraints (for instance, it might predict a slightly negative rainfall amount if not careful, which is physically impossible, though such errors can be fixed by post-processing).

**XGBoost (Extreme Gradient Boosting):** XGBoost is an implementation of the Gradient Boosting framework that has been optimized for efficiency and performance. Gradient Boosting involves building an ensemble of decision trees in a sequential manner. The model starts with a simple initial prediction (like predicting the average outcome for all instances). Then, it iteratively adds decision trees that **fit the residuals** (the errors) of the current model. Each new tree is trained to correct the mistakes made by the previous ensemble. By adding many such trees, the model gradually improves

[mdpi.com](https://www.mdpi.com/2079-9292/13/14/2719#:~:text=The%20framework%20of%20the%20proposed,that%20considers%20diverse%20patterns%20and)

[mdpi.com](https://www.mdpi.com/2079-9292/13/14/2719#:~:text=During%20the%20base%20model%20selection,search%20technique%20determines%20the%20selection)

. The “gradient” part comes from the fact that this residual fitting can be viewed as gradient descent on a loss function. XGBoost specifically introduces regularization terms and clever engineering (like tree pruning, parallelization, and handling missing data internally) that makes the boosting process faster and prevents overfitting

[xgboosting.com](https://xgboosting.com/xgboost-vs-deep-learning/#:~:text=,scores%2C%20aiding%20in%20model%20interpretation)

.

In a weather context, suppose we use XGBoost to predict tomorrow’s temperature. We might supply it with features such as: today’s temperature, humidity, pressure, wind speed, yesterday’s values of those variables (to allow it a bit of memory), perhaps some climatological info (like day of year to account for seasonal baseline), and maybe output from a physics model or other predictors. XGBoost will create a series of decision trees. A single tree might ask questions like “was today’s temperature above 20°C?”; “if yes, was humidity low?”; “if no, what was pressure?” etc., and at the leaves have an adjustment to the forecast. Each tree alone is a weak predictor, but many trees together can model complex nonlinear interactions. Importantly, at each split, XGBoost is essentially selecting the most informative variable to reduce error, which inherently gives a ranking of which features matter. For example, it might find that “pressure tendency” is a very important feature for the next day’s weather, thus using it often in splits. We can extract this information as feature importance scores, which many practitioners do to interpret XGBoost models.

**Strengths of XGBoost:** One big strength is its **interpretability relative to other ML models**. While not as transparent as a single decision tree, the ensemble can still be interpreted through feature importance or partial dependence plots. Practitioners consider XGBoost and similar models easier to trust in sensitive applications because you can at least get a sense of what factors it’s using

[xgboosting.com](https://xgboosting.com/xgboost-vs-deep-learning/#:~:text=,boxes%E2%80%9D%20due%20to%20their%20complexity)

. Additionally, XGBoost is quite **robust and flexible**: it handles outliers by its nature (a few extreme values won’t skew a tree split too much because it chooses splits to reduce overall error), and it can model nonlinear effects and interactions that linear regression cannot. It doesn’t require data normalization and can handle categorical variables (through one-hot encoding or even directly in some implementations). Another strength is **speed**: XGBoost is optimized in C++ and can utilize multiple CPU cores, making training on thousands or even millions of examples feasible. For weather datasets which can be large, this efficiency is valuable. XGBoost also includes regularization parameters (like tree depth limit, L1/L2 penalties on leaf weights) that help it generalize better than an un-regularized boosting approach. This tends to reduce overfitting, which is why XGBoost often performs well on test data in machine learning competitions.

In forecasting, XGBoost can quickly give you a model that captures a lot of signal. For example, it might capture that “if today is much warmer than yesterday, tomorrow will likely also be warmer than climatology” or other intuitive rules, but also less obvious combos like “if humidity is high and pressure fell and it’s spring, then increase rain chance.” It’s essentially doing what forecasters do by heuristics, but in a data-driven, consistent way.

**Limitations of XGBoost:** The main limitation, as mentioned, is that XGBoost doesn’t inherently model time sequences or spatial fields. It treats each training instance independently. We can give it lagged inputs to mimic a memory (e.g., include yesterday’s and day-before-yesterday’s weather as features for today’s forecast), and it will use them if they are predictive. But unlike LSTM, if we wanted to extend it to a longer sequence, we manually have to add those features (yesterday, day-2, day-3, etc.), and at some point that becomes cumbersome or impractical. Essentially, XGBoost has no “state” – it doesn’t remember anything beyond what features you explicitly feed it. So for long-term dependencies, it might miss out unless those dependencies are encoded in the input. For example, an LSTM could in principle remember a pattern from a month ago affecting today; XGBoost would only catch that if you gave it “value from a month ago” as an input feature, which is unusual unless you know to do that.

Another limitation is that while XGBoost is more interpretable than a neural net, it’s still a collection of many decision trees – not something a human can parse directly. One can get a general sense through feature importance (like "pressure drop in the last 3 hours is very predictive of storms"), but you won’t necessarily extract a neat set of human-readable rules easily if there are hundreds of trees. Also, if the problem is extremely complex and data is abundant, deep learning might eventually outperform XGBoost – there’s some evidence that for very large datasets or highly nonlinear problems (like images or speech), tree methods plateau and neural nets keep improving with more data.

**Strengths and Limitations in Weather Forecasting:** For weather forecasting tasks, LSTMs (and RNNs) bring the ability to naturally handle sequential data and potentially model complex atmospheric sequences, but they need a lot of training data and careful design to not overfit or produce physically implausible results. XGBoost brings speed, interpretability, and strong performance on tabular weather features, but may not capture long-sequence dynamics as fully as an LSTM can. In practice, if one has a rich set of features derived from meteorological understanding (including some sequence info like yesterday’s values), XGBoost can be very powerful and sometimes outperform an LSTM, as seen in the wave height case where it edged out the LSTM

[repository.library.noaa.gov](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=and%202017,10.13)

. On the other hand, if the forecasting problem requires recognizing a long-duration phenomenon (like a 30-day oscillation) or an evolving pattern, an LSTM or hybrid might have the edge.

To conclude, LSTM and XGBoost are quite complementary:

* LSTM acts like a **memory-equipped forecaster**, learning the “story” of the weather as it unfolds over time.
* XGBoost is like an **instantaneous analyst**, looking at all available factors at a given time and making a decision based on learned rules.

Understanding these inner workings justifies why combining them could yield an even better model – the hybrid can use XGBoost to analyze complex relationships at each time step or on features the LSTM outputs, while the LSTM ensures that temporal patterns are accounted for

[mdpi.com](https://www.mdpi.com/2079-9292/13/14/2719#:~:text=performance,Lasso%20regression)

. By leveraging LSTM’s sequence learning and XGBoost’s structured data prowess, we aim to harness the best of both methods in weather forecasting. Each alone has limitations, but together, they might overcome many of them – which is exactly the premise we are testing in this study.

**References:**

[dexterenergy.ai](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=Let%E2%80%99s%20start%20with%20the%20%E2%80%98traditional%E2%80%99,surface%20to%20the%20upper%20atmosphere)

[dexterenergy.ai](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=Some%20of%20the%20most%20well,global%20deterministic%20NWP%20models%20are)

Dexter Energy – Article on weather models (deterministic NWP models and examples).

[dexterenergy.ai](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=After%202023%20turned%20out%20to,potential%20benefits%20are%20sometimes%20overstated)

[dexterenergy.ai](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=Intelligence%2FIntegrated%20Forecasting%20System%20,potential%20benefits%20are%20sometimes%20overstated)

Dexter Energy – Notable developments in AI weather models and ECMWF AIFS announcement.

[science.org](https://www.science.org/doi/10.1126/science.adi2336#:~:text=GraphCast%2C%20a%20machine%20learning%E2%80%93based%20method,for%20modeling%20complex%20dynamical%20systems)

Science (Lam et al. 2023) – GraphCast ML model outperforms operational NWP.

[repository.library.noaa.gov](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=modeling%29,Shahidi)

[repository.library.noaa.gov](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=regression%20trees%20%28using%20C5,than%20ANNs%2C%20since%20they%20represent)

Hu et al. (2021) – Early use of neural networks and decision trees in wave forecasting.

[repository.library.noaa.gov](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=note%20that%2C%20as%20with%20ANNs%2C,except%20for%20Bayesian%20networks%2C%20which)

Hu et al. (2021) – Tree models vs ANNs: no physics needed and provide understandable rules.

[ar5iv.org](https://ar5iv.org/pdf/2410.14963#:~:text=combining%20Convolutional%20Neural%20Networks%20,the%20model%27s%20potential%20in%20climate)

Gong et al. (2024) – Hybrid CNN-LSTM improves temperature prediction accuracy.

essay.utwente.nl

Abdul Wahab (2022) – Hybrid (LSTM-XGBoost, CNN-XGBoost) models outperformed single models in load forecasting.

[ijaseit.insightsociety.org](https://ijaseit.insightsociety.org/index.php/ijaseit/article/view/18377#:~:text=Indonesia,model%20was%20better%20since%20it)

Darmawan et al. (2023) – Comparison of GRU and XGBoost for weather prediction (each had advantages in different stages).

[repository.library.noaa.gov](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=and%202017,Lake%20Erie%20yielded%20MAPE%20values)

Hu et al. (2021) – Lake Erie study: XGBoost yielded MAPE ~15.6–22.9% in wave height vs LSTM’s 23.4–30.8%.

[repository.library.noaa.gov](https://repository.library.noaa.gov/view/noaa/53490/noaa_53490_DS1.pdf%20#:~:text=although%20it%20did%20not%20improve,03%20s%20on%201%20CPU)

Hu et al. (2021) – Computational speed: ML models (LSTM 0.24s, XGBoost 0.03s) vs physical model (24h on 60 CPUs).

[nature.com](https://www.nature.com/articles/s41598-024-68906-6#:~:text=Monthly%20climate%20prediction%20using%20deep,one%20month%20time%20step%20ahead)

Nature (Sun & Peng 2020) – Hybrid CNN-LSTM significantly reduced monthly climate prediction error.

[xgboosting.com](https://xgboosting.com/xgboost-vs-deep-learning/#:~:text=,boxes%E2%80%9D%20due%20to%20their%20complexity)

[xgboosting.com](https://xgboosting.com/xgboost-vs-deep-learning/#:~:text=,scores%2C%20aiding%20in%20model%20interpretation)

*XGBoosting.com* – Comparison of XGBoost vs deep learning (interpretability and strengths of XGBoost).

[researchgate.net](https://www.researchgate.net/publication/369179810_An_AI-Enabled_ensemble_method_for_rainfall_forecasting_using_Long-Short_term_memory#:~:text=An%20AI,18%5D%20used)

Yang et al. (2020) via AIMS – LSTM outperforms regression algorithms in daily precipitation forecasting.

[science.org](https://www.science.org/doi/10.1126/science.adi2336#:~:text=resolution%20globally%20in%20under%201,atmospheric%20rivers%2C%20and%20extreme%20temperatures)

Lam et al. (2023) – GraphCast outperforms most accurate operational systems on 90% of metrics.

[dexterenergy.ai](https://dexterenergy.ai/news/weather-models-in-renewable-energy-forecasting/#:~:text=Accurate%20weather%20forecasts%20are%20fundamental,for%20power%20producers%20or%20traders)

Dexter Energy – Importance of accurate weather forecasts for renewable energy integration (reducing volatility and costs).

[resources.eumetrain.org](https://resources.eumetrain.org/data/4/451/english/msg/ver_cont_var/uos3/uos3_ko1.htm#:~:text=The%20MAE%20measures%20the%20average,weighted%20equally%20in%20the%20average)

[resources.eumetrain.org](https://resources.eumetrain.org/data/4/451/english/msg/ver_cont_var/uos3/uos3_ko1.htm#:~:text=The%20RMSE%20is%20a%20quadratic,large%20errors%20are%20particularly%20undesirable)

EUMETRAIN – Definition of MAE and RMSE (error magnitude, linear vs quadratic scoring).

[scribbr.com](https://www.scribbr.com/statistics/coefficient-of-determination/#:~:text=You%20can%20interpret%20the%20coefficient,predicted%20by%20the%20statistical%20model)

Scribbr – Interpretation of R² as variance explained by the model.

[d2l.ai](https://d2l.ai/chapter_recurrent-modern/lstm.html#:~:text=Each%20memory%20cell%20is%20equipped,the%20output%20gate)

[d2l.ai](https://d2l.ai/chapter_recurrent-modern/lstm.html#:~:text=The%20data%20feeding%20into%20the,And%20the%20output%20gate)

Dive into Deep Learning – Explanation of LSTM gating (input, forget, output gates and their roles).

[mdpi.com](https://www.mdpi.com/2079-9292/13/14/2719#:~:text=The%20framework%20of%20the%20proposed,that%20considers%20diverse%20patterns%20and)

[mdpi.com](https://www.mdpi.com/2079-9292/13/14/2719#:~:text=During%20the%20base%20model%20selection,search%20technique%20determines%20the%20selection)

Guo et al. (2023) – Stacking model framework combining XGBoost, LSTM, etc., for improved forecasting (combining strengths of base models).