Weather prediction using deep learning techniques

Dr. David Raj Micheal
Division of Mathematics
School of Advanced Sciences
Vellore Institute of Technology Chennai
Tamil Nadu – 600127
davidraj.micheal@vit.ac.in

S.Akshaya Subramani
Division of Mathematics
School of Advanced Sciences
Vellore Institute of Technology Chennai
Tamil Nadu – 600127
akshaya.s2023c@vitstudent.ac.in

***Abstract*—Weather prediction is essential for a wide range of sectors, including agriculture, transportation, urban planning, and disaster management. Timely and reliable prediction allows for better decision-making, risk management, and resource allocation. This project utilizes deep learning techniques, specifically Recurrent Neural Networks (RNN) to predict weather conditions based on historical data, including date, precipitation, maximum and minimum temperatures, wind speed, and weather type (e.g., drizzle, rain, sun, snow, fog). The RNN model is employed to analyse the sequential nature of weather patterns, learning from past data to predict weather conditions. These models learn the complex interactions between different weather variables and their temporal relationships, enabling more accurate predictions of weather conditions. This approach aims to improve the accuracy and reliability of weather predictions, providing valuable insights for industries that depend on timely and accurate weather information.**

***Index Terms*—Deep Learning techniques, Recurrent Neural Networks (RNN), Sequential Weather Patterns, Decision Support Systems, weather prediction.**

I. Introduction

Weather forecasting is a critical tool that enables better planning and risk management across a variety of fields, from agriculture and urban development to transportation and disaster preparedness. The ability to predict weather patterns reliably and accurately helps these sectors make informed decisions, allocate resources effectively, and prepare for potential disruptions.Conventional weather prediction methods, often based on numerical models, require significant computational resources and rely heavily on physical assumptions, which may limit their accuracy in rapidly changing climates or diverse environments. In recent years, advancements in deep learning have introduced new possibilities for weather forecasting by enabling models to learn directly from vast amounts of historical weather data. This project employs Recurrent Neural Networks (RNNs), a type of deep learning model uniquely designed for analyzing sequential data. RNNs are particularly valuable in weather forecasting because they can capture temporal dependencies—learning from the order and trends in past data to make predictions about weather conditions. By using historical data the RNN model can develop a nuanced understanding of how these variables interact over time. The RNN approach offers an advantage over traditional models by dynamically adapting to data patterns without requiring complex physical equations, which may not always capture real-world variability accurately. With this model, the project aims to deliver reliable forecasts that can support crucial sectors in preparing for weather changes, thereby enhancing safety, operational efficiency, and long-term planning.

II. Objectives

The objective of this project is to develop an effective and reliable weather prediction model using Recurrent Neural Networks (RNNs) to analyse historical weather data. This model aims to accurately forecast weather conditions by capturing temporal dependencies and complex interactions between variables such as precipitation, temperature, and wind speed. By improving forecasting accuracy, the RNN-based approach will support key sectors including agriculture, transportation, urban planning, and disaster management in making informed decisions, optimizing resources, and enhancing preparedness for weather-related challenges.

III. Literature Review

Hewage, P., Trovati, M., Pereira, E., & Behera, A. (2021)This research investigates the performance of lightweight Long Short-Term Memory (LSTM) models in weather forecasting, specifically comparing multi-input multi-output (MIMO) and multi-input single-output (MISO) approaches. The findings reveal that the MIMO-LSTM model, which simultaneously predicts multiple weather parameters, outperforms the complex Weather Research and Forecasting (WRF) model for short-term predictions (up to 12 hours) in terms of accuracy and efficiency. The LSTM architecture effectively learns temporal patterns from historical data, leading to improved forecasting results. While the WRF model excels in long-term forecasting due to its comprehensive climate integration, the deep learning approach is notably more accurate for rain prediction. The study emphasizes the need for larger training datasets and local weather station data to enhance the deep learning model's performance in practical applications. Bochenek, B., & Ustrnul, Z. (2022)Machine learning methods are increasingly vital in weather prediction and climate analysis, as indicated by an analysis of 500 recent scientific articles. Key topics in numerical weather prediction include wind energy, atmospheric physics, and energy processes, while climate research focuses on parametrizations, extreme events, and climate change. Common areas of study encompass wind, precipitation, temperature, pressure, and radiation, utilizing techniques like Artificial Neural Networks, Deep Learning, Random Forest, and Support Vector Machines. Countries such as China, the USA, Australia, India, and Germany lead in this research. The study underscores machine learning's potential in enhancing the accuracy of weather forecasting and climate studies, addressing challenges related to qualitative weather classification, and adapting to advancements in technology and data availability.

Singh, N., Chaturvedi, S., & Akhter, S. (2019)The paper discusses the development of a low-cost, efficient weather forecasting system utilizing machine learning algorithms, specifically random forest classification, on a Raspberry Pi platform. Given the inadequacy of traditional weather prediction methods amidst rapid climate change, the authors aim to create a predictive model that is accessible even in remote areas. The system employs real-time data gathered from various sensors measuring temperature, humidity, and atmospheric pressure to predict rainfall. The proposed application is modular, comprised of three components for data sensing, backend processing, and user interface. With an accuracy rate of 87.90%, the model demonstrates that humidity is the most significant factor in predicting rain. Ren, X., Li, X., Ren, K., Song, J., Xu, Z., Deng, K., & Wang, X. (2021)Deep learning techniques have gained significant attention for weather forecasting due to their ability to efficiently process large meteorological datasets and capture both spatial and temporal patterns. Wang et al. (2021) and Zhang et al. (2020) demonstrated that deep learning models, such as CNNs and RNNs, can enhance weather predictions by overcoming some limitations of traditional Numerical Weather Prediction (NWP) methods, including high computational costs and difficulty in handling complex data. While these methods show great promise, challenges like the need for high-quality datasets, model interpretability, and computational power remain. Despite these obstacles, deep learning is increasingly seen as a valuable supplement to conventional forecasting methods. As research progresses, hybrid models that integrate both approaches are expected to drive future advancements in weather prediction.and real-world data from the fashion retail industry.

Rasp, S., & Thuerey, N. (2021)Recent advancements in deep learning have led to renewed interest in data-driven medium-range weather forecasting, exemplified by the WeatherBench challenge. A convolutional neural network (Resnet) was trained to predict geopotential, temperature, and precipitation up to five days ahead, showing improved forecast skill through pretraining on climate model data. The results indicated that the neural network outperformed previous WeatherBench submissions and was similar to physical baselines, yet still lagged behind high-resolution weather models. The study underscores the balance between specificity and generalization in data-driven methods, revealing that predictions tended to smooth out with longer lead times. Weyn, J. A., Durran, D. R., & Caruana, R. (2020)This study introduces an advanced data-driven global weather forecasting framework that utilizes deep convolutional neural networks (CNN) to predict atmospheric variables on a global scale. The framework incorporates offline volume-conservative mapping to a cubed-sphere grid, enhancements to the CNN architecture, and a loss function that minimizes errors over multiple prediction steps. The developed model yields stable and realistic weather forecasts for several weeks and surpasses existing benchmarks in short to medium-range forecasting, despite being less accurate than high-resolution operational models. The CNN model is trained using ERA5 data and is evaluated against climatology, persistence, and ECMWF models.

Jayasingh, S. K., Mantri, J. K., & Pradhan, S. (2021)The paper discusses the development of hybrid soft computing models for weather prediction, focusing on the challenges posed by climate change and the nonlinear nature of atmospheric phenomena. Traditional statistical methods are being replaced by advanced soft computing techniques, such as Support Vector Machines (SVM), Multi-Layer Perceptron (MLP), and Fuzzy Logic. The authors designed new hybrid models that leverage the strengths of these techniques while mitigating their weaknesses, specifically using weather data from Delhi over five years. The study demonstrates that the proposed hybrid models outperformed conventional methods, yielding better accuracy in forecasting various weather events, including rain and fog. Xhabafti, M. A. L. V. I. N. A., Vika, B., & Sinaj, V. A. L. E. N. T. I. N. A. (2024)The paper evaluates the effectiveness of statistical and deep learning models for rainfall prediction in Albania, crucial for sectors like agriculture significantly impacted by climate change. Utilizing data from January 1901 to December 2022, the study compares the Auto-Regressive Integrated Moving Average (ARIMA) and Error, Trend & Seasonal (ETS) models against Long Short-Term Memory Network (LSTM) and Deep Feed Forward Neural Network (DFNN) models. The study finds that the ETS model outperforms ARIMA, while DFNN shows superior performance compared to LSTM. Ultimately, the ETS model exhibited the lowest Root Mean Square Error (RMSE), thereby providing the most accurate rainfall forecasts. The research emphasizes the importance of reliable rainfall forecasting for sustainable agricultural practices in Albania, suggesting further exploration of hybrid modeling approaches for improved accuracy.

Chattopadhyay, A., Nabizadeh, E., & Hassanzadeh, P. (2020)The study presents a data-driven framework for predicting extreme weather events, specifically heat waves and cold spells in North America, utilizing analog forecasting and deep learning methods. It highlights the superiority of Capsule Neural Networks (CapsNets) over traditional methods, achieving prediction accuracies between 40.6% and 82.0% depending on lead times and seasons. Incorporating surface temperature data alongside midtropospheric circulation patterns significantly enhances prediction accuracy, reaching up to 88.2%. The framework demonstrates the effectiveness of multivariate approaches for timely extreme weather forecasts, which could improve early warning systems. CapsNets, due to their advanced feature extraction capabilities, outperformed ConvNet architectures, particularly in predicting events up to 5 days in advance.

IV. Methodology

A. Data Structure

The dataset used in this study was sourced from Kaggle (https://www.kaggle.com/), one of the most well-known and regarded platforms for hosting datasets pertaining to various fields.

- The dataset consists of time-series data with each entry representing a specific timestamp
- Each entry contains several columns such as:
  - **Date:** Timestamp of the observation.
  - **Weather-related features:** `temp_max` (maximum temperature), `temp_min` (minimum temperature), `precipitation`, `wind`, and `weather` (categorical weather conditions like sunny, cloudy, etc.).

B. Data Preprocessing

- **Missing Data Handling:** Check for any missing values within the dataset and handle them appropriately, either by filling with mean/median values or by dropping missing rows.
- **Duplicate Removal:** Identify and remove any duplicate records in the dataset to maintain data integrity.
- **Feature Selection:** Identify relevant features for the prediction model. For simplicity, one feature (e.g., temp_max) is selected as the target for prediction.
- **Time-Series Data Preparation:** Convert the data into sequences where each input consists of the past n time steps and the output is the value for the next time step.

C. Feature Selection

The sliding window technique is used to generate features for the model:

- **Target Variable Selection:** In this code, the target variable is chosen from the dataset by selecting the third column (df.iloc[:, 2:3]). This column will be used to both predict future values and as the feature for training the model. Essentially, the model is set up to predict the values of this column based on its past values.
- **Feature Variable Creation:** The features are derived from the same target variable (training set). Specifically, the past values of this column are used to predict the current value.
- For each data point, the previous 10 values (time_steps = 10) of the target variable are used as features. This is known as the sliding window approach, where a sequence of past observations is used to predict the next value.

D. Feature Creation for LSTM

- The sliding window technique creates a series of features, each representing the past 10 time steps. Each new data point will have 10 previous values as its features.
- This is an implicit form of feature selection because you're not choosing specific columns or external features but instead using a historical window of the same feature.

E. Data Splitting

The dataset is split into training and testing sets:

- **Training Set:** A portion of the data (typically 80%) is used for training the model.
- **Testing Set:** The remaining data (typically 20%) is used to evaluate the model's performance.
  This is done to ensure the model generalizes well to unseen data and is not overfitting to the training data.

F. Model Compilation

- **Optimizer:** The model is compiled using the Adam optimizer, which adjusts learning rates during training to optimize performance.
- **Loss Function:** A loss function, such as Mean Squared Error (MSE), is used for regression tasks to calculate the difference between predicted and actual values. The goal is to minimize this loss during training.

G. Building the model

- **Model architecture:** A Recurrent Neural Network model is used in this project.
  - **Input Layer:** The input layer is defined to accept sequences of historical data (e.g., 10 time steps).
  - **LSTM Layers:** Use Long Short-Term Memory (LSTM) layers to capture dependencies in the time-series data. These layers are designed to process sequential data, learning patterns over time.
  - **Dropout Layers:** Dropout layers are added to prevent overfitting by randomly setting some layer outputs to zero during training
  - **Output Layer:** The output layer is typically a dense layer with one neuron to predict a continuous value (e.g., next day's temperature).

H. Training the Model

- **Fitting the Model:** The model is trained on the training dataset for a specified number of epochs. During training, the model learns to minimize the loss function by adjusting its weights based on the input data.
- **Batch Size:** The batch size determines how many samples are processed before the model's internal parameters are updated. Smaller batch sizes can lead to more stable convergence.

I. Model Fitting Process

- During each epoch, the model learns by adjusting its weights based on the loss function. It tries to minimize the error between the predicted and actual target values (temperature, in this case).
- The validation set is used to check how well the model generalizes to new data, and its performance (validation loss) is logged during training.

V. RECURRENT NEURAL NETWORK(RNN) ARCHITECTURE

Recurrent Neural Networks (RNNs) are a type of neural network specifically designed to handle sequential data. Unlike traditional feedforward neural networks, RNNs have a unique architecture that allows information to persist, making them ideal for tasks where context from previous time steps influences predictions. This makes RNNs highly effective for time series analysis, natural language processing (NLP), and other sequential tasks.

A. Core Components of LSTM Architecture

- **Input Layer:** The input layer consists of input vectors that represent the sequential data. For instance, in natural language processing, each word in a sentence can be represented as a vector (using techniques like word embeddings).
- **Hidden Layer(s):** The hidden layer is where the core of the RNN's processing occurs. It contains recurrent units that maintain a hidden state over time. The hidden state captures information from previous time steps, allowing the RNN to remember past inputs. The hidden layer can have multiple layers (stacked RNNs) to increase the model's capacity.
- **Output Layer:** The output layer generates predictions based on the current hidden state. Depending on the task, this could involve producing a single output (e.g., predicting the next word) or multiple outputs (e.g., class probabilities for each time step).
- **Recurrent Connections:** RNNs have connections that loop back from the hidden layer to itself. This allows the network to pass information from one time step to the next, enabling it to maintain a form of memory.
- **Recurrent Connections:** RNNs have connections that loop back from the hidden layer to itself. This allows the network to pass information from one time step to the next, enabling it to maintain a form of memory.
  - **Tanh:** Often used to keep the hidden state values between -1 and 1, helping with gradient flow.
  - **ReLU:** Sometimes used in hidden layers, but care must be taken to avoid issues like exploding gradients.

B. How RNNs function Work

- **Sequential Processing:** RNNs process data one element at a time, maintaining a hidden state that captures information from previous time steps. For example, in a sentence, the RNN can remember the context of earlier words to predict the next word.
- **Backpropagation Through Time (BPTT):** This is a variant of backpropagation used to train RNNs. It involves unfolding the RNN through time and calculating gradients for each time step, allowing the model to learn from the entire sequence.

C. Benefits of RNNs

- **Memory of Previous Inputs:** RNNs can remember previous inputs, making them suitable for tasks where context is crucial.
- **Parameter Sharing:** RNNs use the same parameters across different time steps, reducing the complexity of the model.
- **Flexibility:** RNNs can handle variable-length input and output sequences, making them versatile for various applications.

D. Common Uses of LSTMs

- **Natural Language Processing:** Tasks such as language modeling, text generation, and machine translation.
- **Speech Recognition:** Converting spoken language into text by processing audio signals sequentially.
- **Time Series Prediction:** Forecasting future values based on historical data, such as stock prices or weather patterns.
- **Sentiment Analysis:** Classifying the sentiment of text data, such as determining if a review is positive or negative.

E. Challenges and Limitations

- **Vanishing and Exploding Gradients:** RNNs can struggle with long sequences due to these gradient issues, which can hinder learning.
- **Training Complexity:** Training RNNs can be computationally intensive and time-consuming, especially for long sequences.

F. Variants of RNNs

- **Long Short-Term Memory (LSTM):** A type of RNN designed to overcome the vanishing gradient problem by using memory cells and gates to control the flow of information.
- **Gated Recurrent Unit (GRU):** A simpler variant of LSTM that combines the forget and input gates into a single update gate, making it computationally more efficient.
- **Bidirectional RNNs:** These networks process data in both forward and backward directions, improving context understanding.

VI. MODEL EVALUATION

Post-training, each model's performance was rigorously evaluated using a range of metrics:

- **Mean Squared Error (MSE):** MSE measures the average of the squares of the errors, which indicates the average squared difference between predicted and actual values. A lower MSE indicates better model performance.
- **Mean Absolute Error (MAE):** MAE represents the average absolute difference between predicted and actual values. It provides a straightforward measure of prediction accuracy, with lower values indicating better performance.

- **Root Mean Squared Error (RMSE):** RMSE is the square root of MSE and provides an error metric in the same units as the predicted values. It is sensitive to outliers and gives a higher weight to larger errors, making it useful for understanding the model's performance in practical terms.
- **R-squared (R^2):** R^2 indicates the proportion of variance in the dependent variable that can be explained by the independent variables in the model. An R^2 value close to 1 suggests that the model explains a significant portion of the variance, indicating strong predictive capability.

VII. Analysis of Result

The model appears to perform well based on the metrics provided. The MAE and RMSE values suggest that the average prediction error is relatively low, while the R^2 value indicates a strong explanatory power. The close values of MAE and RMSE suggest that there are no extreme outliers significantly affecting the model's performance. However, it is always good to visually inspect the residuals to confirm this. The R-squared (R^2) value of 0.8711 signifies that approximately 87.11% of the variability in the target variable is explained by the model, reflecting its strong explanatory power.

The above result shows the metrics and their values:

- **MSE:** 2.1835
- **MAE:** 7.8103
- **MAE:** 7.8103
- **RMSE:** 2.7947

VIII. Conclusion

In conclusion, the RNN model exhibits strong predictive performance, demonstrating a reliable level of accuracy in its predictions. The evaluation metrics indicate that the model maintains a moderate level of error while effectively capturing the underlying patterns in the data. Additionally, the model explains a significant portion of the variance in the target variable, highlighting its effectiveness. Overall, the RNN model serves as a valuable tool for making accurate predictions, with opportunities for further refinement and enhancement through additional tuning and analysis of residuals.

IX. References

References

[1] P. Hewage, M. Trovati, E. Pereira, and A. Behera, "Deep learning-based effective fine-grained weather forecasting model," *Pattern Analysis and Applications*, vol. 24, no. 1, pp. 343-366, 2021.

[2] N. Singh, S. Chaturvedi, and S. Akhter, "Weather forecasting using machine learning algorithm," in *2019 International Conference on Signal Processing and Communication (ICSC)*, Mar. 2019, pp. 171-174, IEEE.

[3] B. Bochenek and Z. Ustrnul, "Machine learning in weather prediction and climate analyses—applications and perspectives," *Atmosphere*, vol. 13, no. 2, p. 180, 2022.

[4] X. Ren, X. Li, K. Ren, J. Song, Z. Xu, K. Deng, and X. Wang, "Deep learning-based weather prediction: a survey," *Big Data Research*, vol. 23, p. 100178, 2021.

[5] S. Rasp and N. Thuerey, "Data-driven medium-range weather prediction with a resnet pretrained on climate simulations: A new model for weather bench," *Journal of Advances in Modeling Earth Systems*, vol. 13, no. 2, e2020MS002405, 2021.

[6] J. A. Weyn, D. R. Durran, and R. Caruana, "Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere," *Journal of Advances in Modeling Earth Systems*, vol. 12, no. 9, e2020MS002109, 2020.

[7] S. K. Jayasingh, J. K. Mantri, and S. Pradhan, "Weather prediction using hybrid soft computing models," in *Intelligent Systems: Proceedings of ICMIB 2020*, Springer Singapore, 2021, pp. 35-47.

[8] M. A. Xhabafti, B. Vika, and V. A. Sinaj, "A comparative study of statistical and deep learning model-based weather prediction in Albania," *WSEAS Transactions on Computer Research*, vol. 12, pp. 151-160, 2024.

[9] A. Chattopadhyay, E. Nabizadeh, and P. Hassanzadeh, "Analog forecasting of extreme-causing weather patterns using deep learning," *Journal of Advances in Modeling Earth Systems*, vol. 12, no. 2, e2019MS001958, 2020.