Weather Prediction using RNN

Aarushi Prabhu Nagpure
axn220015@utdallas.edu

Rubina Parveen
rxl220014@utdallas.edu

Deep Padmani
dmp210005@utdallas.edu

Harsh Rashmikant Patel
hxp230001@utdallas.edu

Krittika Paul
kxp230001@utdallas.edu

Gayatri Mangire
grm210001@utdallas.edu

Abstract*—Today in society, accurate weather forecasts are esessential across numerous sectors like agriculture, industry, transportation, and daily activities. With the ongoing ramifications of climate change, precise weather prediction becomes crucial for seamless mobility and safe day-to-day operations. Our project employs Python and its libraries to develop an RNN-based model that utilizes a comprehensive dataset with features such as date, precipitation, temperature, wind, and weather conditions to predict the maximum temperature for the subsequent day in Seattle. Utilizing machine learning and deep learning methodologies, we seek to advance weather prediction capabilities and facilitate informed decision-making and enhancing the resilience in the face of evolving climate patterns.

Keywords—Weather Forecasting, RNN, Machine Learning

I. Introduction

Predicting the weather is a critical component of modern life, influencing both everyday plans and long-term strategic decisions for individuals and organizations alike. These forecasts, provided by weather applications, are crucial for daily activities. Weather forecasting involves the application of science and technology to estimate atmospheric conditions at a specific location and time. People have been trying to predict the weather for a long time, but things got more interesting in the 1800s with the development of more sophisticated scientific techniques.

The importance of weather forecasting extends beyond convenience; it is vital for protecting lives and property, enhancing public health and safety, and bolstering economic resilience and quality of life. By predicting weather patterns accurately, forecasts help mitigate weather-related losses and promote societal welfare.

In this project, we will employ Recurrent Neural Networks (RNNs) to predict temperatures using collected data. RNN, is a type of deep learning model, particularly suited for time-series data analysis and applications like speech recognition. Unlike conventional neural networks, RNNs utilize their internal state (memory) to process sequences of data, enabling them to predict future events based on historical information. This project aims to harness the predictive power of RNNs to forecast future temperatures effectively.

II. Problem Definition

The primary aim of this project is to construct a machine-learning model utilizing Recurrent Neural Networks (RNNs) for precise temperature prediction. Forecasting weather plays a crucial role across multiple industries, yet it presents significant challenges due to the intricate behavior of atmospheric conditions. This project analyzes historical weather data and makes predictions to facilitate informed decision-making in critical areas such as agriculture, transportation, and emergency management.

The project contains several fundamental elements: collection and preprocessing of data, developing, and training the machine learning model, and assessing the model's forecasting accuracy using relevant metrics. The project's success will be measured based on the model's ability to generate accurate temperature predictions within a specified time frame compared to ground truth observations.

III. Algorithm And Techniques Used

A. Algorithm Description

The project tries to implement a simplified form of Recurrent Neural Network (RNN) to forecast maximum temperatures based on historical weather data. Recurrent Neural Networks (RNNs) are a class of neural networks which are designed to process sequential data by maintaining memory of previous inputs. When an RNN processes a sequence, it uses the information it has learned from previous data points to influence the output it produces.The states are updated throughout the learning process. This is particularly useful for timestamp prediction where the goal is to forecast future values based on previously observed values in a time series.

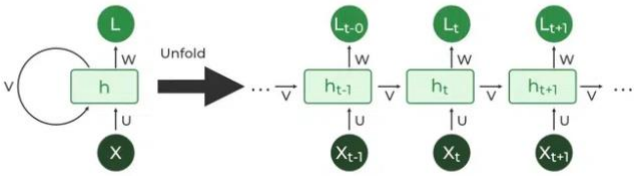

Figure 1: Recurrent Neural Network

B. Data Description

The dataset analyzed in this study represents meteorological data from Seattle, Washington, USA, spanning from 2012 to 2015. It includes daily measurement of various weather related variables:

- Date: It ranges from the year 2012 to the year 2015.
- Precipitation: Recorded in millimeters, this variable measures the amount of rain, sleet, hail or snow that during a given day.

- Maximum Temperature: Highest temperature recorded on each day in degree Celsius.
- Minimum Temperature: Lowest temperature recorded on each day in degree Celsius.
- Wind Speed: Speed of wind recorded in meters per second.
- Weather Type: This is a categorical variable that describes a day's general weather condition like sunny, rainy.

The dataset provides a comprehensive overview of the climatic treads and variations in Seattle over the specified four year period, offering valuable insights into seasonal changes and day to day fluctuations.

	date	precipitation	temp_max	temp_min	wind	weather	tmax_tomorrow
0	01-01-2012	0.0	12.8	5.0	4.7	0.11	10.6
1	02-01-2012	10.9	10.6	2.8	4.5	0.58	11.7
2	03-01-2012	0.8	11.7	7.2	2.3	0.58	12.2
3	04-01-2012	20.3	12.2	5.6	4.7	0.58	8.9
4	05-01-2012	1.3	8.9	2.8	6.1	0.58	4.4
...	...	...	...	...	...	...	...
1456	27-12-2015	8.6	4.4	1.7	2.9	0.58	5.0
1457	28-12-2015	1.5	5.0	1.7	1.3	0.58	7.2
1458	29-12-2015	0.0	7.2	0.6	2.6	0.08	5.6
1459	30-12-2015	0.0	5.6	-1.0	3.4	0.00	5.6
1460	31-12-2015	0.0	5.6	-2.1	3.5	0.00	5.6

Figure 2: Seattle Weather Dataset

C. *Data Preprocessing and Feature Extraction*

A target variable column is added to this dataset named as the 'tmax_tomorrow' column. This target variable, derived from the temp_max column, represents the maximum temperature forecasted for the following day. It is calculated by shifting the daily maximum temperatures forward, aligning each day's temperature with the subsequent day's forecast.

A thorough exploratory analysis of the dataset confirmed the robustness of the dataset, which is complete with no missing entries. The five-number summary of the dataset provides a detailed quantitative breakdown.

index	date	precipitation	temp_max	temp_min	wind	tmax_tomorrow
count	1461	1461.0	1461.0	1461.0	1461.0	1461.0
mean	2013-12-31 00:00:00	3.02943189596167	16.43908281998631	8.234770704996578	3.241136208076666	16.434154688569475
min	2012-01-01 00:00:00	0.0	-1.6	-7.1	0.4	-1.6
25%	2012-12-31 00:00:00	0.0	10.6	4.4	2.2	10.6
50%	2013-12-31 00:00:00	0.0	15.6	8.3	3.0	15.6
75%	2014-12-31 00:00:00	2.8	22.2	12.2	4.0	22.2
max	2015-12-31 00:00:00	55.9	35.6	18.3	9.5	35.6
std	NaN	6.680194322314738	7.349758097360177	5.023004179961266	1.4378250588746193	7.35461208763703

Figure 3: Statistical description of the dataset

The range of each variable is well defined through there minimum values, first quartiles, medians, third quartiles and maximum value. The count for every variable is 1461 which means that there are no null values in the dataset.

After adding the 'tmax_tomorrow' column, a detailed pre-processing routine was performed.

- We conducted a thorough check for any null values in the dataset.
- The 'date' variable was converted to datetime format to facilitate easy analysis. Subsequently, we extracted and stored the year, month, and day as separate columns to analyze any potential seasonal and temporal effects on weather prediction.
- The categorical variable 'weather' was removed from the dataset. This decision was made considering the focus of the analysis on quantitative variables that have a more direct impact for numerical modeling.
- To understand the interdependencies between variables, a correlation heatmap was generated. This visualization gives insights into how various factors are related to the target variable 'tmax_tomorrow'.

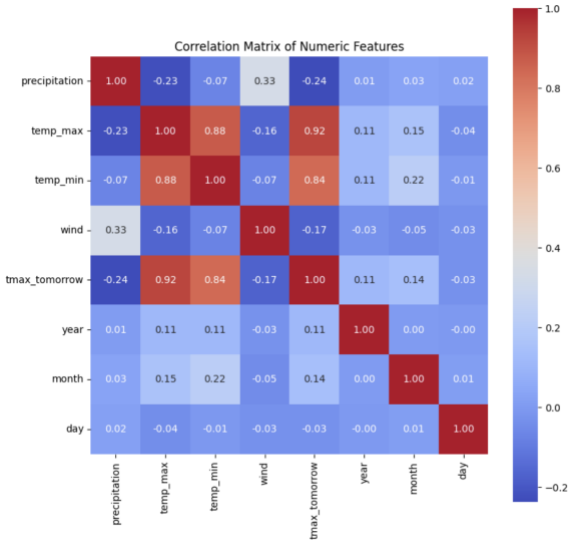

Figure 4: Heatmap depicting the correlation between the columns in the dataset.

The key insights from the correlation matrix can be summarized as below:

- Strong correlations: There is a significant correlation between 'tmax' and 'tmin'.
- Moderate positive correlation: The 'month' and 'year' exhibit a modest but positive correlation with 'tmax_tomorrow'.
- Descent correlation: Variables 'day', 'precipitation' and 'wind' show a minimal correlation. It indicates that they have less contribution in predicting the next day's maximum temperature.

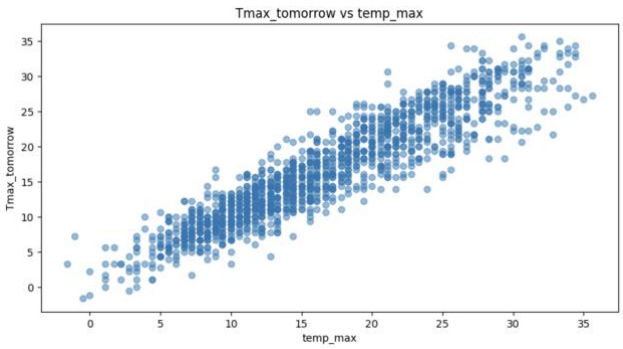

Figure 5.1: Correlation between tmax_tomorrow and temp_max

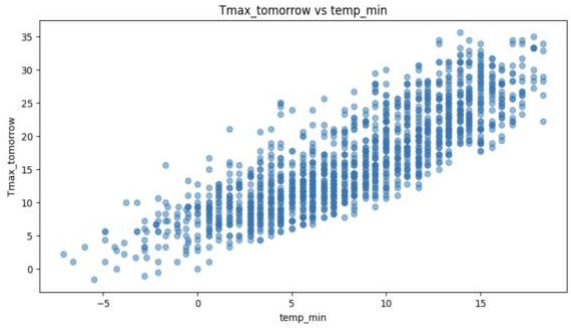

Figure 5.2: Correlation between tmax_tomorrow and temp_min

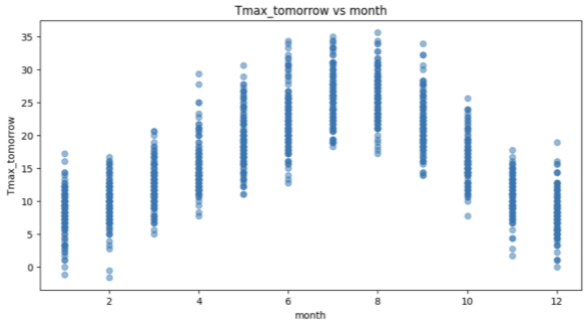

Figure 5.3: Correlation between tmax_tomorrow and month

Following a thorough examination of the correlations between the various columns of the dataset, the features that were selected for the input of the predictive model were: 'precipitation', 'temp_max', 'temp_min', 'wind', 'year', 'month', 'day'. These features include both the meteorological and temporal variables, providing a comprehensive set of predictors that encompass weather conditions and time – based trends.

D. *Data Normalization and Splitting*

To ensure consistency in measurement scale across all input features, normalization was applied using 'StandardScalar' from the sklearn.preprocessing package. This normalization process adjusts each feature to have zero mean and unit variance, effectively standardizing the data distribution.

To split the data into training and testing subsets, the data was split in an 80:20 ratio, allocating 80% data for training and 20% for testing. The split provides a substantial amount of data for model learning while reserving a representative portion for testing to access model generalizability.

E. *Model Implementation*

The architecture of RNN is structured into three primary components: input layer, hidden layer and output layer. The input layer is configured to accept the features extracted – precipitation, maximum and minimum temperature, wind speed, year, month and day. There is one hidden layer with 14 nodes and the output layer comprises of a single node that predicts the maximum temperature for subsequent day.

During the forward pass the RNN processes the input sequence one time step at a time, updating its hidden state at each step. The hidden state is initialized to zero or a small random value. At each time step, the input is passed through the input layer and combined with the previous hidden state to produce a new hidden state. The hidden state is then used to generate the output prediction at the current time step. The hidden state from the current time step becomes the input to the next time step, allowing the network to maintain memory of past inputs.

Tanh function is used in 'forwardPass' function, where it is applied to the hidden state computation. It squashes the input values to the range [-1, 1] and introduces non-linearity, allowing the network to learn complex patterns. It is a common choice for activation functions in RNNs and other neural network architectures.

The tanh (hyperbolic tangent) function is defined as:

$$f(x) = \frac{\left(e^{x} - e^{-x}\right)}{\left(e^{x} + e^{-x}\right)}$$

or equivalently:

$$f(x) = tanh(x) = \frac{2}{1+e^{-2x}} - 1$$

During training, the RNN's parameters (weights and biases) are updated to minimize the difference between predicted outputs and the actual value. This is achieved through backpropagation through time (BPTT). In the backward pass, the loss is calculated as difference between predicted outputs and actual values, which is computed using a loss function, such as Mean Squared Error (MSE) or Mean Absolute Error (MAE). The gradients of the loss function with respect to the parameters of the network are computed using the chain rule of calculus. The parameters of the network (weights and biases) are updated using gradient descent optimization. Specifically, here a basic form of gradient descent with a fixed learning rate is used.

A bunch of libraries such as pandas, numpy, scikit-learn, are used to achieve the tasks of data pre-processing, feature engineering and implementation of the RNN. Libraries like matplotlib, seaborn are used to visualize the results of the predictive model.

IV. Results and Discussions

A. Mean Square Error (MSE): Mean sqaured error is calculated by simply taking the average of the squared difference between true values and predicted values in the dataset. It is a measure of the variance of the residuals.

$$MSE = (1/n) * \Sigma (original - predictions)^2$$

B. Mean Absolute Error (MAE): Mean Absolute Error is calculated by taking the average of the absolute difference between true values and predicted values in the dataset. This measures the average of the residuals in the dataset.

$$MSE = (1/n) * \Sigma |original - predictions|$$

C. *R-Squared Fit:* R- squared is also known as the coefficient of deteremination. It is a measure of how well a model fits the given dataset in other words, it measures how close the regression line is to the true values. R^2 is a scale-free score and lies between 0 and 1 (where 0 indicates that the model doesn't fit the dataset at all, and 1 indicates that the model fits the dataset perfectly).

$$R^2 = 1 - (SSE/SST)$$

where,

SSE (Sum of Squares Error) = Σ(original – predictions)^2

SST (Total Sum of Squares) = $\sum(yi - y')^2$

(here, y_i is the observed value, y' is the mean of the observed values)

D. *Loss:* The loss function is also known as the Cost Function which is a mathematical measure of error. The target of machine learning is to reduce the loss function thereby improving the performance of the model.

$$Loss = (1/2m) * \Sigma (original - predictions)^2$$

Metric	Training Score	Testing Score
Mean Square Error (MSE)	0.1345	0.1621
Mean Absolute Error (MAE)	0.2910	0.3171
R^2 Fit	0.8606	0.8415
Loss	0.1355	0.1642

Fig. 6: Summary of Performance Metrics on Test and Train Data

The above performance metric indicates that model is performing decently well in predicting the output (tmax_tomorrow) given the input features. The R^2 value for training data is 0.8606 and for test data is 0.8415 which is very close to 1 and indicates that the model fits well with the given dataset. On the other hand, lower values of MSE, MAE, and loss underscores the model's proficiency in minimizing errors and improving accuracy in predicting new data.

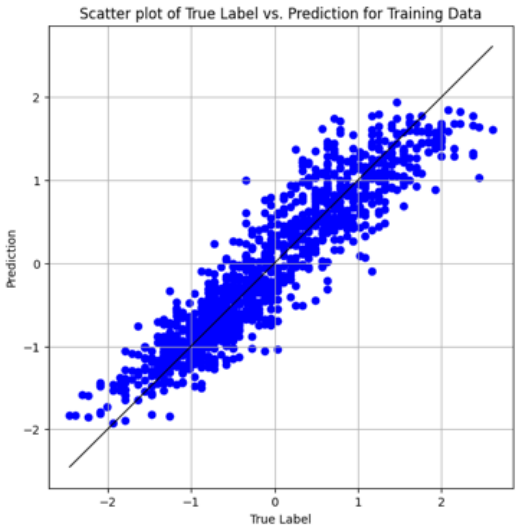

Fig 7: Predicted vs True value plot for training data

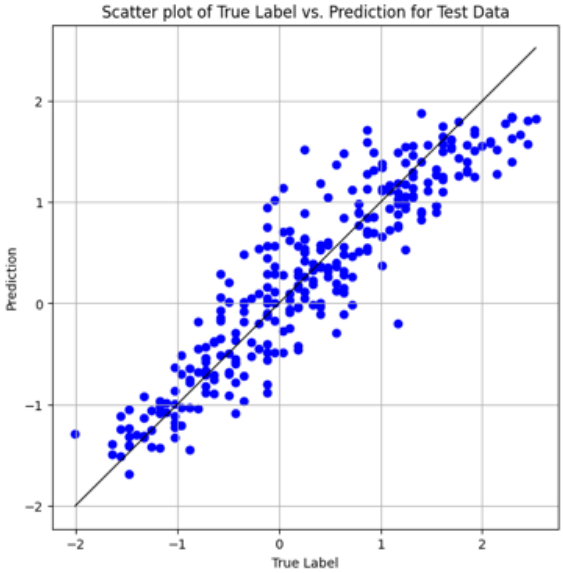

Fig 8: Predicted vs True value plot for testing data

In figures 7 and 8, we observe the correlation between the predicted and true values for the Training and Test Datasets, respectively. Notably, the predicted data points, marked in blue, are closely aligned with the diagonal line, which represents the true value in each plot. This proximity of the predicted points to the diagonal line indicates that the model is in line with the observations . This alignment thereby proves the ability of the model to capture the key data patterns in both test and train data thereby outputting correct predictions in both the scenarios.

V. Conclusions

In conclusion, the Recurrent Neural Network (RNN) model that has been employed for predicting tomorrow's maximum temperature (tmax_tomorrow) based on various other weather attributes has demonstrated strong performance across multiple evaluation metrics.

The model seems to have a pretty good accuracy as the value of Mean Square Error (MSE), Mean Absolute Error (MAE), and Loss are considerably for the Training and Testing Datasets. Finally, it is worthy to note that, also, the R-squared (R^2) Fit scores are close to 1, which means that the variations of the target value are reliably captured by the model. Visualizing the model's predictions against the real values corroborated that they are quite accurate. The scatter charts with the correlation graph between actual and forecasted values show the clear trend of data points near the diagonal line, where an excellent matching of observation and prediction is holding.

Overall, the RNN approach has recently became one of the superior methods as it allows to generate output values, resulting in more realistic approximation. It is one of the most flexible technologies that allows to be deployed in different scenarios. The model has the capability to resist the different testing conditions leading to low error rates which bring out the relevance of the model in the different fields such as agriculture, energy management and emergency preparedness, where precision in weather forecasting is indispensable for making strategic decisions as well as risk mitigation.

REFERENCES

[1] A. G. Salman, B. Kanigoro and Y. Heryadi, "Weather forecasting using deep learning techniques," 2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Depok, Indonesia, 2015, pp. 281-285, doi: 10.1109/ICACSIS.2015.7415154.

[2] A. Kumar, P. Siddhi, K. U. Singh, T. Singh, D. P. Yadav and T. Choudhury, "Exploring Advanced Deep Learning Techniques for Reliable Weather Forecasting," 2023 World Conference on Communication & Computing (WCONF), RAIPUR, India, 2023, pp. 1-7, doi: 10.1109/WCONF58270.2023.10235085. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[3] K. G, Y. Sushmitha, K. L. Saranya, P. Naga Ramya Sri and P. Amulya, "Rainfall Prediction Using Deep Learning and Machine Learning Techniques," 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2023, pp. 1-7, doi: 10.1109/ACCAI58221.2023.10199905.

[4] Jun Zhang and K. F. Man, "Time series prediction using RNN in multi-dimension embedding phase space," SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218), San Diego, CA, USA, 1998, pp. 1868-1873 vol.2, doi: 10.1109/ICSMC.1998.728168.

[5] Jingyang Wang, Xiaolei Li, Jiazheng Li, Qiuhong Sun, Haiyao Wang, "NGCU: A New RNN Model for Time-Series Data Prediction”, Big Data Research,Volume 27,2022,100296,ISSN 2214-5796, doi.org/10.1016/j.bdr.2021.100296.

[6] Salman, Afan & Kanigoro, Bayu & Heryadi, Yaya. (2015). Weather forecasting using deep learning techniques. 281-285. 10.1109/ICACSIS.2015.7415154.

[7] Rather, Akhter Mohiuddin, Arun Agarwal, and V. N. Sastry. 2015. “Recurrent Neural Network and a Hybrid Model for Prediction of Stock Returns.” Expert Systems with Applications 42(6):3234–41.

[8] Kök, Ibrahim, Mehmet Ulvi Şimşek, and Suat Özdemir. 2018. “A Deep Learning Model for Air Quality Prediction in Smart Cities.” International Conference on Big Data (BIGDATA) A 1983–90.

[9] Divya Chauhan and Jawahar Thakur. Data mining techniques for weather prediction: A review. International Journal on Recent and Innovation Trends in Computing and Communication, 2(8):2184–2189, 2014.

[10] Aditya Grover, Ashish Kapoor, and Eric Horvitz. A deep hybrid model for weather forecasting. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 379–386. ACM, 2015.