Forecasting weather and its different components using KNN & RNN

Costa, Piu Terasa
CSE, American International
University-Bangladesh
Dhaka,Bangladesh
piucosta98@gmail.com

Zafrin Sultana
CSE, American International
University -Bangladesh
Dhaka,Bangladesh
zafrinjolly123@gmail.com

Rimon Nath
CSE, American International
University-Bangladesh
Dhaka,Bangladesh
rimonnath11@gmail.com

Abstract: The weather has a big influence on people's health and happiness. Allergen concentrations and high levels of pollution have been associated to fluctuations in birth rates and sperm counts, as well as outbreaks of pneumonia, influenza, bronchitis, and other morbidity consequences. As the environment changes at such a rapid rate these days, traditional weather forecasting methods are becoming less efficient and time-consuming. Also, weather depends on so many factors like temperature, pressure, humidity, wind direction, speed and many more. Better and more dependable weather forecasting systems are needed to overcome these concerns. These projections have an influence on a country's economy and people's lives. Developing a weather prediction system considering various factors mentioned above that can be used in remote areas is the main motivation of this work. This model will predict advanced updates of weather considering different fields and factors by using different machine learning methods.

Keywords: *Machine Learning, Weather Prediction, Weather Forecasting, Recurrent Neural Network, K Nearest Neighbor.*

I. INTRODUCTION

Natural catastrophes that occur regularly have impacted negatively our planet, putting people's lives in danger. A natural catastrophe is a sudden occurrence, such as an accident or natural calamity, that results in significant damage or several deaths. Over the last several years, a large number of catastrophic calamities have occurred all over the world. Many fatalities, disability, and shelter damage were reported, as well as tragedies of villages being ripped apart. Rainfall patterns have shifted in many regions, leading to greater floods, or heavy rain, as well as more frequent and severe heat waves. Oceans and glaciers have also changed oceans are warming and getting more acidic, ice caps are melting, and sea levels are rising. As The climate is changing at a drastic rate nowadays, which makes the old weather prediction methods less effective and more hectic due to different indicators factor. And as predictions affect a nation's economy and the lives of people, improved and reliable weather prediction methods are required. Weather forecasting and prediction include gathering and communicating information about future weather conditions based on meteorological measurements. Climate predictions with a lead time of more than two weeks are referred to as long-range or extended-range forecasts.

Computer models are used by meteorologists to forecast the weather. And processing power has advanced significantly. The capacity to predict the weather over a few days is still restricted by three factors: the amount of data available, the time available to study it, and the complexity of weather occurrences. Other than that, the nature and reliability of communication systems available for forecast dissemination and the makeup and requirements of the user community also create problems in predicting the weather perfectly. And the models are a bit expensive too.

In LSTM, training takes longer, requires more memory, and is simple to overfit. While GRU models still have issues with slowdown convergence and low learning efficiency, resulting in excessive training time and even under-fitting.

The main goal of this project is to create a low-cost, dependable, and effective weather forecasting program in Python utilizing the machine learning concepts of KNN and RNN. KNN was used for weather prediction and finding its accuracy. An RNN is used for the prediction of an individual parameter. By this, we also can predict long-term Forecasting trends of weather and its different parameter that can be used in remote areas as well. A complicated memory module such as the Gated Recurrent Unit (GRU) or Long Short-Term Memory is not required here.

II. BACKGROUND STUDY

A. *Machine Learning*

Machine learning (ML) may be a kind of AI (AI) that permits software programs to enhance their prediction accuracy without being expressly designed to try to do so. Machine learning algorithms use prior data as input to estimate future output values. Machine Literacy is a pivotal part of the fleetly expanding discipline of data wisdom. Algorithms are trained to induce groups or prognostications using statistical approaches, revealing pivotal perceptivity in a data mining enterprise. Following that, this perceptivity drive decision-making within operations and enterprises, with the thing of impacting important growth KPIs. As big data expands and grows, the demand for data scientists will rise, challenging their backing in relating the most important business questions and, as a result, the data demanded to answer them. Machine learning classifiers are divided into three groups.

- *Supervised Machine Learning:* The use of labeled datasets to train algorithms that duly classify data or prognosticate issues is appertained to as supervised machine literacy. As further data is introduced into the model, the weights are acclimated until the model is well fitted. This happens throughout the cross-validation process to corroborate that the model doesn't overfit or underfit. Neural networks, nave bayes, linear regression, logistic regression, random forest, support vector machine (SVM), and other approaches are used in supervised learning.

- *Unsupervised Machine Leaning:* Unsupervised machine learning analyzes and clusters unlabeled datasets using machine learning methods. Without the need for mortal commerce, these algorithms uncover hidden patterns or data groupings. Because of its capacity to find parallels and contrasts in data, it's perfect for exploratory data analysis, cross-selling techniques, consumer segmentation, picture and pattern recognition. Principal component analysis (PCA) and singular value decomposition (SVD) are two typical methodologies for reducing the number of features in a model through the dimensionality reduction process. Neural networks, k- means clustering, probabilistic clustering approaches, and other algorithms are applied in unsupervised learning.

- *Semi-supervised Leaning:* Between supervised and unsupervised learning, semi-supervised learning is a good compromise. During training, it uses a lower labeled data set to guide classification and attribute extraction from a larger, unlabeled data set. Semi-supervised learning can break the problem of having not enough labeled data (or not being suitable to go to label enough data) to train a supervised learning algorithm.

B. Weather Prediction

Weather forecasting is the operation of current technology and knowledge to forecast the state of the atmosphere for a coming time and a given place. The many weather forecast methods are as follows:

- *Synoptic weather prediction:* It's the classic system of rainfall soothsaying. The term "synoptic" refers to the observation of numerous meteorological factors at the same time. A meteorological center creates a series of synoptic maps every day to keep track of the changing weather, which serve as the foundation for weather forecasts. It entails collecting and anatomizing a large measure of experimental data from thousands of rainfall stations.

- *Numerical weather prediction:* It forecasts the weather with the help of a computer. Supercomputers execute complex computer algorithms that anticipate a variety of meteorological conditions. The equations applied aren't accurate, which is one drawback. However, the forecast will not be fully correct, If the weather's first stage is not completely understood.

- *Statistical weather prediction:* They are applied in confluence with numerical approaches. It makes use of weather data from the history, assuming that the future would be analogous to the history. The introductory thing is to determine which factors of weather are good predictors of coming circumstances.

  This approach can only be used to forecast the general weather.

III. LITERATURE REVIEW

The section provides a review of the literature on RNN and KNN prediction methods as well as other machine learning approaches which is targeted to be used for the statistical weather prediction.

A. K Nearest Neighbor (KNN)

The KNN algorithm is a simple supervised machine learning technique that may be used to handle classification and regression problems. It's simple to set up and comprehend, but it has the disadvantage of becoming noticeably slower as the amount of data in use grows. The key benefit of KNN over other algorithms is that it can be used to classify several classes.

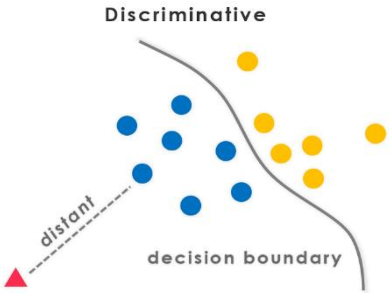


Figure 1: The figure of Discriminative Technique with KNN

KNN is a good method to use if the data has more than two labels or if you need to classify the data into more than two categories. Because it represents the conditional probability of a sample belonging to a specific class, KNN is a discriminative algorithm. Consider how one comes to the KNNs decision rule to see what is meant. Discriminative Classifiers determine which features in the input are most useful in discriminating between different classes. The K Nearest Neighbor (KNN) method is a fundamental deterministic locating technique used in fingerprinting. By using the correct selection algorithm, the KNN's performance may be considerably improved.

B. Recurrent Neural Network (RNN)

RNNs are a type of neural network that may be used to represent a series of events. RNNs, which are formed from feedforward networks, function similarly to human brains. Simply said, recurrent neural networks are better at anticipating sequential input than other algorithms. RNNs are a type of neural network that is both strong and robust, and because they are the only ones with internal memory, they are one of the most promising algorithms now in use. "Whenever a sequence of data exists, the temporal dynamics that link the data are more important than the spatial content of each individual frame." Lex Fridman is a writer who specializes in science fiction and fantasy fiction (MIT). The information in a feed-forward neural network flows in just one direction: from the input layer to the output layer, passing through the hidden layers. The data travels across the network in a straight line, never passing through the same node twice.

Feed-forward neural networks have no recollection of the data they receive and are poor predictors of what will happen next. A feed-forward network has no sense of order in time because it just considers the current input. Except for its training, it has no recollection of anything that transpired in the past.
An RNN's information is looped back on itself. It evaluates the current input as well as what it's learned from prior inputs before making a decision.

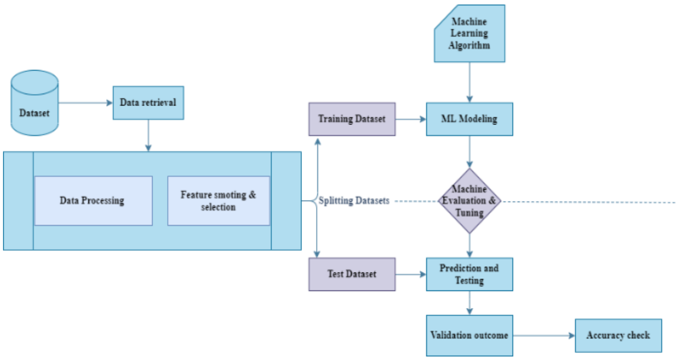
IV. Methodology

The whole prediction of weather is evaluated by using the K-nearest neighbor (KNN) model. In addition, Recurrent Neural Network (RNN) model is implemented to individual parameters of weather (humidity, temperature, air pressure, and so on) to forecast the long-term trend of the parameters.

The process of developing a machine learning model can be complicated, and the model that is developed must be constructed in such a manner that it fits the problem perfectly. To determine if a model will do a good job of predicting the target on new and future data, the model should always be evaluated. Because future instances have unknown target values, it's needed to check the accuracy metric of the ML model on data for which we already know the target answer, and use this assessment as a proxy for predictive accuracy on future data.

The backend is responsible for the following:

- Fetching data either from the dataset.
- Pre-processing the raw weather data.
- Split into training and test part.
- Predicting the expected weather using KNN.
- Predicting the components of weather individually using simple RNN model.
- Evaluate the whole model.



A. Dataset

The dataset contains historical weather characteristics (temperature, pressure, and relative humidity) for major North American cities and other places across the world from 2012 to 2017. Hourly data points are collected, totaling approximately 45000 data points. The full dataset was taken from Kaggle. The collection of data comprises 5 years of high temporal resolution (hourly readings) data on a variety of meteorological variables, including temperature, humidity, air pressure, and so on. This information is accessible for 30 cities in the United States and Canada, as well as six cities in Israel.

1) The feature variables

- Date
- Humidity
- Pressure
- Temperature
- Wind direction
- Wind speed

Based on these feature variables, the model predicts the weather description and find the accuracy.

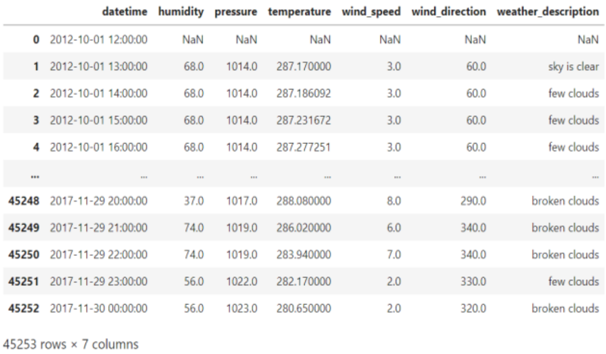
1) The target variable:

The target variable of a dataset is the feature of a dataset about which we want to gain a deeper understanding. For the dataset, the target variable is: Wind description.

B. Data Processing

A very important step before applying any model is preprocessing. It is an important step as we cannot work with raw data because it won't result good enough as it might have faulty values. So, the quality of the data should be checked before applying machine learning or data mining algorithms. For data processing here used predefined libraries pandas, Keras, and sklearn for pre-processing.

1. For the model, the BOSTON city is considered for predicting weather and merging all feature variables only for Boston City.



	datetime	humidity	pressure	temperature	wind_speed	wind_direction	weather_description
0	2012-10-01 12:00:00	NaN	NaN	NaN	NaN	NaN	NaN
1	2012-10-01 13:00:00	68.0	1014.0	287.170000	3.0	60.0	sky is clear
2	2012-10-01 14:00:00	68.0	1014.0	287.186092	3.0	60.0	few clouds
3	2012-10-01 15:00:00	68.0	1014.0	287.231672	3.0	60.0	few clouds
4	2012-10-01 16:00:00	68.0	1014.0	287.277251	3.0	60.0	few clouds
...	...	...	...	...	...	...	...
45248	2017-11-29 20:00:00	37.0	1017.0	288.080000	8.0	290.0	broken clouds
45249	2017-11-29 21:00:00	74.0	1019.0	286.020000	6.0	340.0	broken clouds
45250	2017-11-29 22:00:00	74.0	1019.0	283.940000	7.0	340.0	broken clouds
45251	2017-11-29 23:00:00	56.0	1022.0	282.170000	2.0	330.0	few clouds
45252	2017-11-30 00:00:00	56.0	1023.0	280.650000	2.0	320.0	broken clouds

45253 rows × 7 columns

2. The rows with NULL values are filled with fillna function and the used method is bfill. bfill() is used to backward fill the missing values in the dataset. It will backward fill the NaN values that are present in the pandas dataframe. Hence, the NULL values were filled.

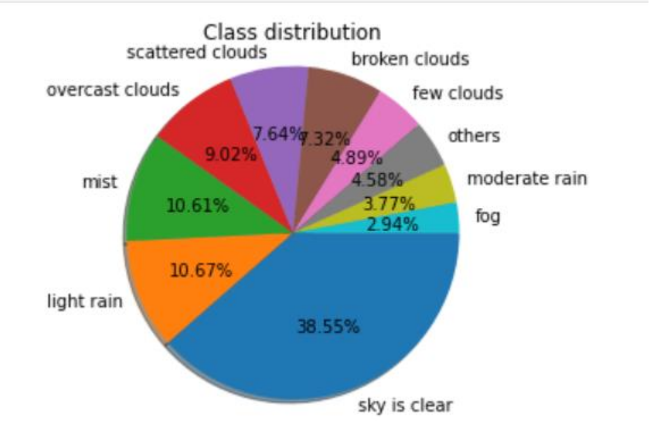
   Another method that is used to adjust the wind speed and direction data is median (). The median () method calculates the median (middle value) of the given data set. This method also sorts the data in ascending order before calculating the median.

3. After analyzing the data, some least times arrived classes for weather description were most likely useless and so removed, using smoting will balance the distribution. It balances class distribution by randomly increasing minority class examples by replicating them. This algorithm helps to overcome the overfitting problem posed by random oversampling.

C. Data Visualization

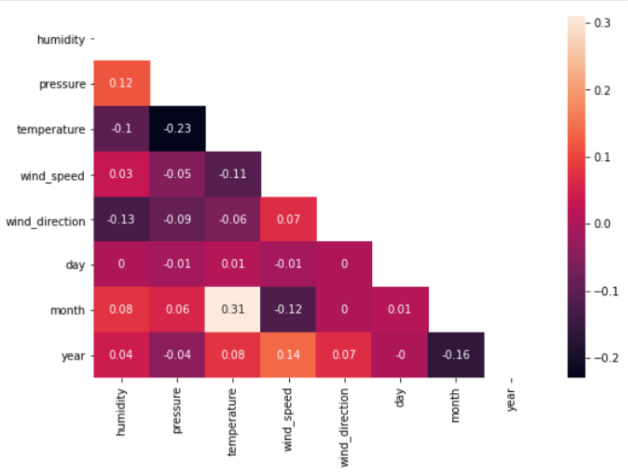
1) Visualization in pie chart

Different classes of weather description is shown below through pie chart where most of the time (38.55%) the weather was "clear sky" for Boston city. Then with 10.67% appearance, the weather was "light rain" for Boston. The least arrived weather for Boston city is "fog" with 2.94% appearance.



2) Visualizing The Correlation Matrix

Visualizing The Correlation Matrix for the weather parameters of Boston city. A correlation matrix is simply a table which displays the correlation coefficients for different variables. This matrix consists of rows and columns that show the variables. Each cell in a table contains the correlation coefficient.

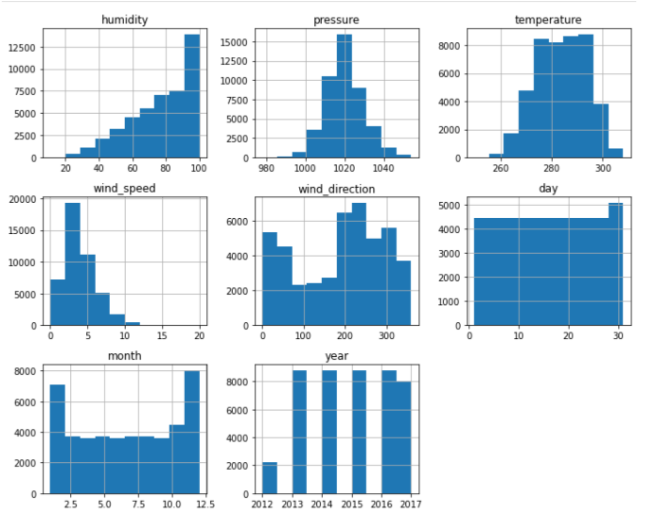


As seen in the correlation matrix, none of the features possess strong co-dependency, hence we do not need to drop any of the features and we can use all of them to train our machine in order to forecast the weather.

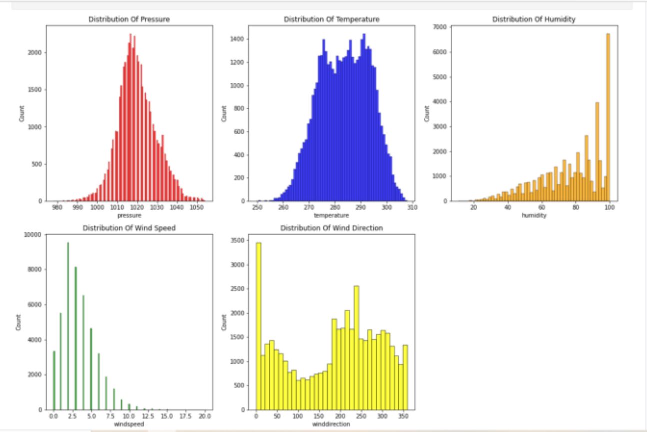
Then Histogram represents all attributes and Visualization of all attributes without class attributes.

3) Showing Histogram Of the parameters

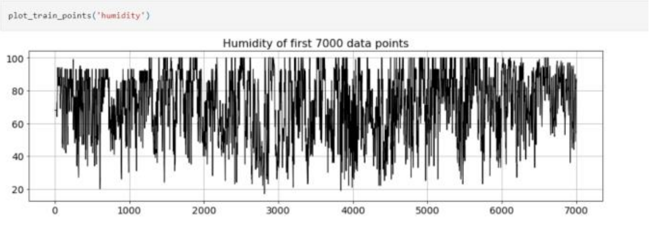
Histograms for different parameters is visualized with hist () method. The graphical representation of parameters (humidity, pressure, temperature, wind speed, wind direction, day, month and year) organizes the group of data points into a specified range. Histograms visualize quantitative data or numerical data.



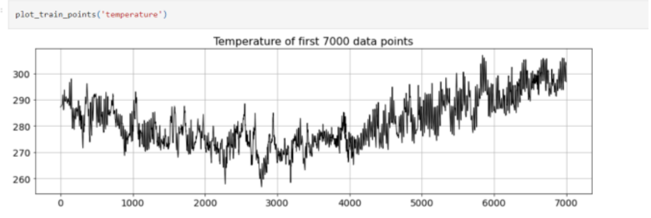
4) Visualization to check the distribution of the attributes



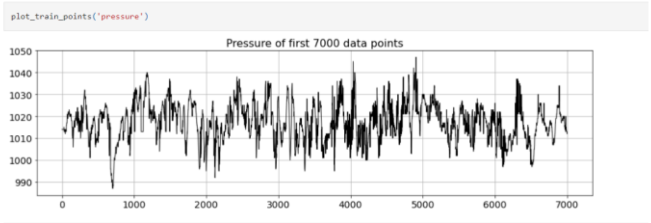
5) Visualization of Individual parameters for the first 7000 data.



```
plot_train_points('humidity')
```



```
plot_train_points('temperature')
```



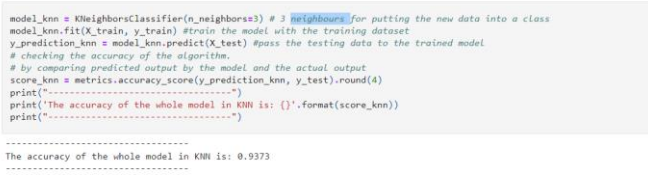
```
plot_train_points('pressure')
```

V. Result of implementation

To assess how effectively our machine learning models work, the dataset was divided into train and test sets. The train set is used to fit the model, and the train set's statistics are well-known. The test data set is the second collection, and it is only utilized to make predictions.

Result of KNN

After splitting the well processed dataset into train and test data, KNN model was implemented to predict and evaluate the accuracy of the prediction. The accuracy was actually good with almost 94%.

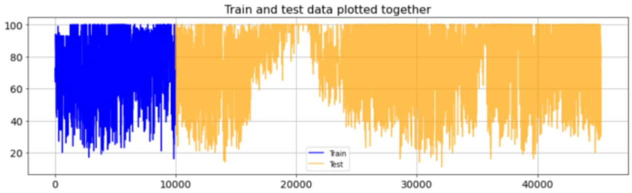


```
model_knn = KNeighborsClassifier(n_neighbors=3) # 3 neighbours for putting the new data into a class
model_knn.fit(X_train, y_train) #train the model with the training dataset
y_prediction_knn = model_knn.predict(X_test) #pass the testing data to the trained model
# checking the accuracy of the algorithm.
# by comparing predicted output by the model and the actual output
score_knn = metrics.accuracy_score(y_prediction_knn, y_test).round(4)
print("----------------------------------")
print('The accuracy of the whole model in KNN is: {}'.format(score_knn))
print("----------------------------------")
```

```
----------------------------------
The accuracy of the whole model in KNN is: 0.9373
----------------------------------
```

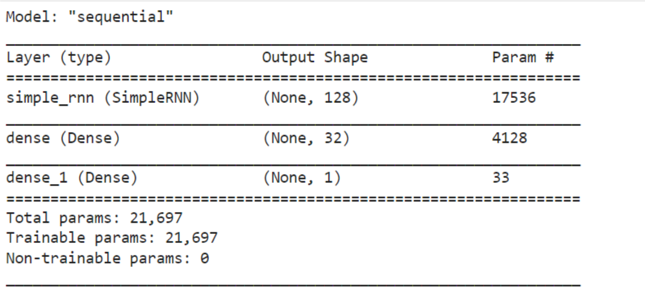
Result of RNN

To implement the RNN model, dataset was split into train and test set where the training data value was 7000 and the rest was considered as test data so the model accuracy could be known precisely.



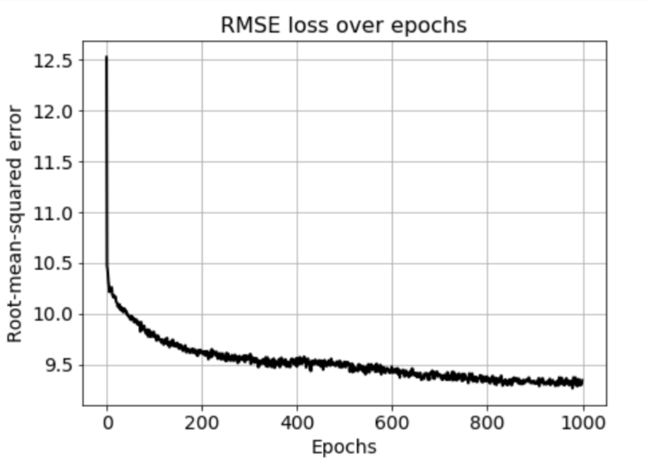
Then the train and test data were converted into matrix with the step value=8 and then modelled for applying RNN. We built a simple function to define the RNN model. It uses a single neuron for the output layer because we are predicting a real-valued number here. As activation, it uses the ReLU function. Following arguments are supported.

- neurons in the RNN layer
- embedding length (i.e. the step length we chose=4)
- neurons in the densely connected layer=32
- learning rate=0.001

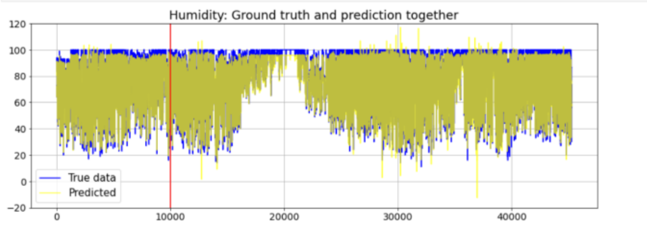


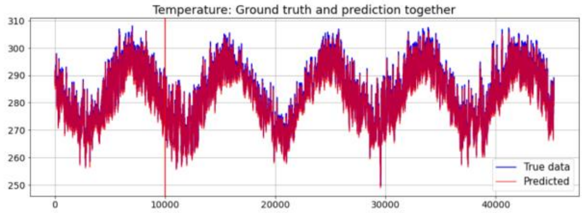
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
simple_rnn (SimpleRNN)       (None, 128)               17536
_________________________________________________________________
dense (Dense)                (None, 32)                4128
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 33
=================================================================
Total params: 21,697
Trainable params: 21,697
Non-trainable params: 0
_________________________________________________________________
```

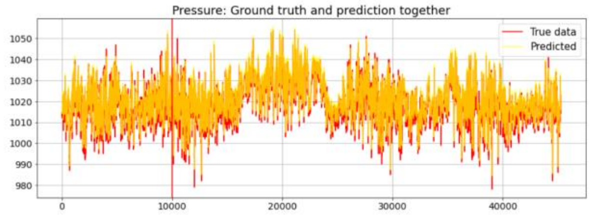
Loss was calculated using root mean square error. The root means square error (RMSE) is the residuals' standard deviation (prediction errors). Residuals are a measure of how far the data points are from the regression line; RMSE is a measure of how spread out the residuals are. To put it another way, it shows how closely the data is grouped around the line of greatest fit.



After compiling the RNN model, the model was used to predict the different parameter like temperature, pressure, humidity. The test and predicted data were visualized one over another so that comparison becomes easy. First, the test data was plotted and above that the RNN models predicted values were plotted. And it provided an incredible result.







The model is able to predict sudden increase and decrease in temperature, humidity and pressure. There was no indication of such shape or pattern of the data in the training set (the boundary is denoted by the vertical red line), yet, it is able to predict the general shape pretty well from the first 7000 data points.

VI. Discussion

To properly execute the models, a pre-processed sample of data is held out that has been labelled with the target (weather description) from the training data source. Once the model is trained, the model is sent to the held-out observations for which the target values are known. Then the predictions that are returned by the ML model are compared against the known target value. Finally, the summary matrix tells that how well the predicted and true values matched.

VII. CONCLUSION

Weather forecasting is vital in our everyday life, particularly in agriculture and associated diligence. We created and applied a general computational architecture based on learning models in this study. Using the machine learning ideas of KNN and RNN, we developed a low-cost, reliable, and successful weather forecasting in Python. KNN was applied to determine the delicacy of weather prediction. For the prediction of a single parameter, an RNN is applied. We can also estimate long-term weather forecasting patterns and their numerous parameters, which may be employed in remote places

We created a strategy for generating dependable weather prediction from general, publicly available data using RNN and KNN in this work. As can be seen from the preceding explanation, the proposed models give excellent outcomes when compared to other algorithm prediction strategies. Despite the fact that we only displayed the findings for one megacity, the model is capable of forecasting all metropolises. We can also observe that both algorithms have more than 90-accuracy rate. Because of the increased perfection, we can additionally use pressure, temperature, humidity, weather descriptions, and the whole state of the wind data set. We may use it to forecast the weather in any position or country. It may also be used for climate forecasting, predicting the weather on a seasonal to interannual time frame. Eventually, we may state the code was examined at and tested. The entire system was performed in accordance with the project's planned purpose and pretensions, according to the test findings. With a high level of perfection and effectiveness, the weather forecasting software program was suitable to anticipate the weather for the following seven days to within a one- degree month.

VIII. REFERNCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] 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.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.