Abstract

Numerical Weather Prediction (NWP) require a significant amount of computational power to solve complicated mathematical equations in order to provide a prediction based on current and historical weather conditions. We present a lightweight data-driven weather forecasting model based on Artificial Neural Networks in this study. Time-series weather data , investigated using the most recent application of ANN i.e,
Long Short-Term Memory (LSTM) layered model, which is a specialised version of Recurrent Neural Network (RNN) for weather prediction. The goal of this model is to create and test a short-term weather forecasting model based on LSTM. The proposed deep learning model comprises of stacked LSTM layers that forecast weather using surface weather characteristics over a specific time period. The model is tested with various numbers of LSTM layers, optimisers and optimised for effective short-term weather predictions.

1.Introduction

The technique of predicting the condition of the atmosphere based on certain time frames and places is referred to as weather forecasting. In recent years, unusual weather, such as cold temperatures, heavy snow, and heavy rain, have resulted in human injury and death, property damage, and health and environmental issues. In other words, climate change has a direct detrimental influence on daily living quality. Unexpectedly high temperatures, in particular, endanger outdoor workers. Temperature predictions can help people decide what to wear on a particular day and schedule their outside activities. Depending on the temperature forecast uses, an appropriate model should be built for a given location or region with at least a 7-day projection.

Meteorological institutions often forecast prediction outcomes using Numerical Weather Prediction(NWP), model which use computer algorithms to offer a forecast based on current weather conditions by solving massive systems of non-linear mathematical equations based on physical models. However, NWP frequently generates skewed temperatures that grow in proportion to the local elevation and topographical complexity. Furthermore, forecasting spatial and temporal temperature changes in locations with complicated topography remains a difficulty for NWP models.

2. Data & Methodology

For monitoring and forecasting purposes, surface weather parameters are observed and reported. The surface parameters such as wind direction, wind speed, humidity, pressure, etc are considered.

2.1 Data Collection and preparation

Various meteorological parameters are acquired from era5 for the period of 1 January 2010 to 31 December 2020. This is the training dataset that is used to train the suggested models. Similarly, a dataset for the year 2021 has been developed to test the network. The December 2021 dataset is also retrieved to serve as the ground truth for evaluating the whole model. To evaluate the entire model, the model is run in forecast mode using the same CSV data format for 2021. The training data set has been normalised to keep each value between 0 and 1, and the testing and evaluation data sets have been normalised using the same maximum and minimum variable values. Then apply the sliding window method for each dataset based on seven days data to train and next 4 hours data as the model's output.

2.2 Feature selection

Feature selection by information gain is simple baseline approach. Here information . gain is the measure of the dependency between the variables. It is equal to zero if and only if two variables are independent, and higher values mean higher dependency. For this project we were able to get 15 parameters from the ERA5 reanalysis dataset. To have an effective and lightweight model we will be finding the dependency between target column i.e, 2m-temperature and all other 14 parameters to select top 50 percentile.

Dew point temperature	0.582	Total column rainwater	0.124
Relative humidity	0.458	Large scale rainrate	0.105
Surface pressure	0.354	Wind speed	0.098
Total precipitation	0.252	Total column snowwater	0.087
Volumetric soil water	0.224	Temp of snow layer	0.061
Wind direction	0.203	Snow fall	0.042
Evoparation	0.201	Soil type	0.019

Table 1: The Mutual Information gain between all parameters

2.3 Selection of Hyperparameters of LSTM

As weather is a chaotic process, a variety of elements play critical roles in regulating the change of a weather variable. Similarly, a neural network's deep structure works as extremely complicated logic; hence, modifying network variables influences prediction outcomes. The learning rate, batch size, number of epochs, regularisation, weight initialization, number of hidden layers, number of nodes, and other network characteristics are referred to as neural network hyper parameters.

Activation functions for hidden layers decide whether the input feature is deemed to be related to the output nodes.

In general, the activation functions must be constrained. When the value of an activation function exceeds a given bound, it is said to be activated. It is eliminated if it is smaller than the bound. To concentrate the suggested LSTM based prediction model on weather data, particularly temperature, we must employ a variety of activation functions, including linear, sigmoid, and ReLU functions. To that purpose, each of these three activation functions is used in this model, with one chosen for the greatest prediction accuracy.

2.4 Data Preprocessing

First, we import the necessary libraries and load the dataset file into pandas, and then we verify basic statistical information about the dataset using Python's in-built describe function. Finally, we do some exploratory data analysis on our dataset, such as outlier discovery and treatment. Then we divided the dataset into train and test, dividing them on the basis of an 90-10 ratio, although in the case of date/year datasets, it would be better to separate on the basis of year/month (data before 2020 goes into train, data after 2020 goes into test). Following that, we determined whether the dataset had any categorical columns and encoded them in order to feed the model. Then we scale the data into the (0,1) range to increase the likelihood of avoiding bias problems between features and to reduce computational power, then we divide the dataset into slots to get the desired shape of data to feed into LSTM.

2.5 Proposed model

we define the model with parameters, and finally we compile it to make a temperature prediction. To draw a graph after forecasting the data, we utilise the root mean squared error approach to evaluate the model performance on train and test data.Several different configurations have been utilised to train and test the proposed models. Figure 4 depicts the general architecture of the proposed model. Each layer consists of a number of nodes. Optimizers are generally used in deep learning networks to minimise a given cost function by updating the model parameters such as weights and bias values. For better learning, it is often useful to reduce learning rates as the training progresses when training a deep network. This can be achieved by using adaptive learning rate methods. Therefore, the fixed learning rate and adaptive learning rate methods have been explored to train the proposed deep models .

The proposed model is evaluated using two different types, namely multi-input multi-output (MIMO) and multi-input single-output (MISO). In the MIMO, all 7 variables are fed into the network, which are expected to predict the same 7 variables as the output. In contrast, in the MISO approach, all 7 variables are fed into the network with a single variable output. In MIMO, only one model is required for weather forecasting involving 7 different variables. Whereas, in MISO, 7 different models are required as each of them is trained to predict a particular weather parameter with all 7 variables as input.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 7, 100)            113200

 dropout (Dropout)           (None, 7, 100)            0

 lstm_1 (LSTM)               (None, 7, 200)            240800

 dropout_1 (Dropout)         (None, 7, 200)            0

 lstm_2 (LSTM)               (None, 100)               120400

 dense (Dense)               (None, 26)                2626

=================================================================
Total params: 477,026
Trainable params: 477,026
Non-trainable params: 0
_________________________________________________________________
```

fig 1: Model architecture

3. Results and Discussion

3.1 LSTM performance over a single location

For initial testing of the model we predicted weather for 1 location. i.e, Kalyani. This testing is carried out with multiple epochs, different optimizers. The below graphs, show that Adam is better than SGD After trying multiple different variations the optimal solution here was 125 epoche a batch size of 32, adam optimizer.

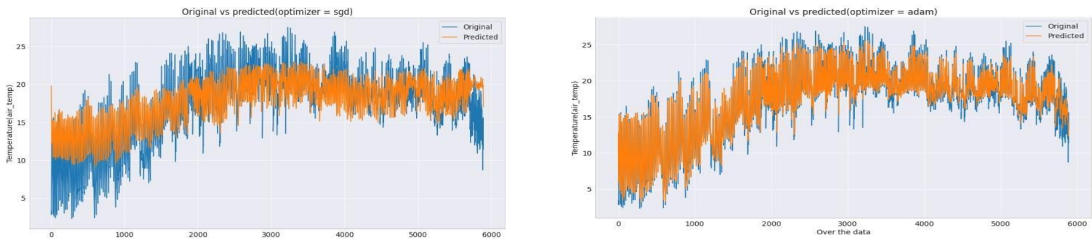

fig 2&3: The graph of original values vs model predicted values

Adam Optimiser	Earlier Model	Current Model
Train RMSE	0.033	0.043
Test RMSE	1.577	0.04
R2 Score	0.8	0.9

Table 2: Current model vs Previous model

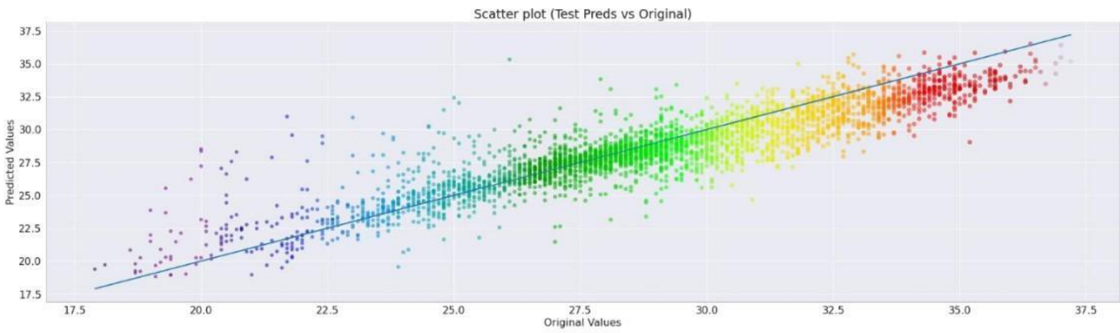

fig 4: Scatter plot for test predictions vs original values

3.2 LSTM performance over multiple locations

To test the model accuracy for the entire India but it would be difficult to evaluate the model accuracy if we consider the whole India and predict the continuous temperature values. So for testing the accuracy of the model, we will be selecting 8 random places in India and predicting the temperatures over a year.

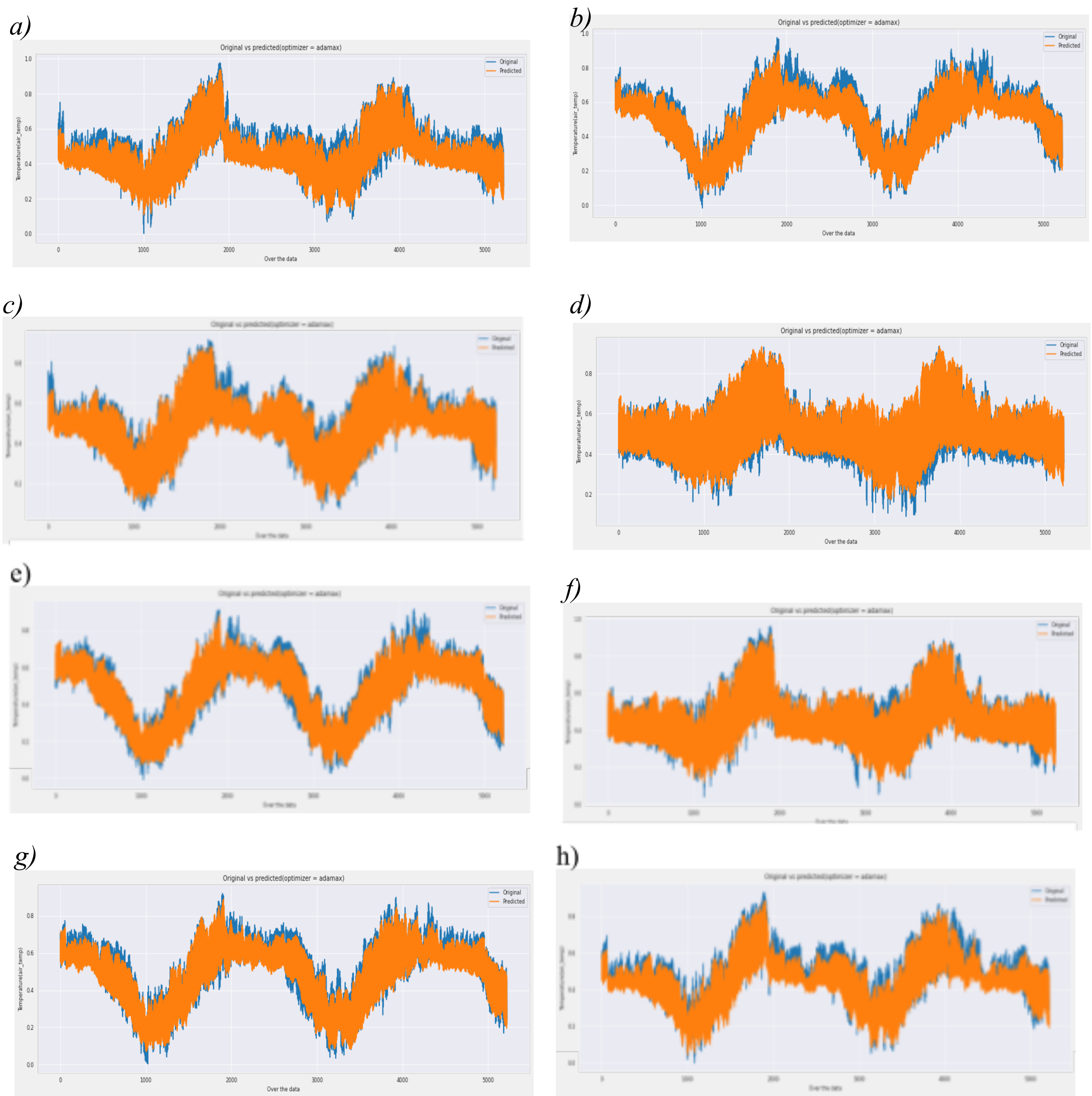

fig 5: Set of 8 locations predicted vs original values

4. Conclusions

4.1 Summary

This research has predicted air temperature using Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM). The temperature prediction model consists of three processes. The first process is a pre-process consisting of interpolation, feature extraction, data normalisation , splitting of dataset. The second process is the training process using RNN and LSTM. The third process is testing. The results of this study indicate that the use of RNN and LSTM can be used for daily temperature prediction. The test was carried out in two different ways. Firstly we tested the model with two optimization models namely SGD and Adam for single location temperature predictions and Next we used the model to forecast the temperatures of 26 randomly selected locations in India. The results obtained using Adam's optimization model for the second case with 125 epochs in the training data yielded an accuracy of 94.79%, and for the test data, the accuracy produced 89.36%. Predictions using data from the past twenty years have gotten better results. Sharing data with 90:10 also got us better accuracy. Therefore, it can be concluded that the optimization model, the amount of data, and data sharing can affect the results obtained.

4.2 Future scope

This suggested model has the potential to be much better in the future with a few updates, such as forecasting temperatures for India as continuous data. Meteorological stations typically collect various types of climate-related data from all parts of India via local stations, airports, and other established locations; however, the problem with this type of data collection is that it is easy to collect this type of data in urban and semi-urban areas, whereas it is not possible to set up the equipment for data collection in every part of India, and most rural areas are not developed enough to have a station for data collection. So, in order to address these issues, instead of predicting temperature for discrete points in India, the proposed model in this report can be used with a modified input, i.e., instead of predicting temperature for discrete points in India, we can take data from available large meteorological stations as input, train the model with past 20 years data, and get a forecast for India as a continuous output. And, rather than dealing with a large amount of numerical data, since we will be forecasting for India as a continuous body, it would be simple to adapt the model such that it accepts input as images and outputs predictions as a colour coded map of India.