Weather Evaluation using Deep Learning

Pritish Samant
Hochschule Hamm-Lippstadt
Deep Learning
Lippstadt, Germany
pritish-sanjay.samant@stud.hshl.de

***Abstract*—Deep Learning is one of the essential subdomains in technology. Many unsolved problems can be solved using deep Learning. In this paper, we will explore whether evaluation and the techniques. As well as get to know about Deep Learning and simple Recurrent Neural Networks, an algorithm of Deep Learning. We will also see the implementation of the Prediction of weather and its evaluation using the deep learning algorithm Recurrent Neural Network.**

***Index Terms*—Deep Learning, Weather Evaluation, Prediction, Recurrent Neural Network**

I. Introduction

Weather Forecasting is essential for predicting various climate-related changes, storms, snow, rain, and natural calamities. Though weather is very dynamic and cannot be predicted hundred percent correctly, it will help us in many ways. By using higher-order numerical methods appropriate for solving model equations at high-resolution discrete elements, climate science and weather communities' ongoing work continuously improves weather models' fidelity [1]. Theoretically, Prediction is nothing but the probability of an event occurring in the future based on current parameters. Thus, it is possible to define the probability of an event occurring in t year, i.e., in t year, and its calculation formula is:

$$P = \frac{n}{t} \times 100\%$$

[2].

In recent years, technology has surpassed its expectations. New tech fields are emerging at an incredible pace, and so is Deep Learning. Deep Learning is about learning different parameters on its own and predicting based on its Learning. Deep Learning can predict things based on the provided data set. Likewise, It is possible to evaluate weather conditions using Deep Learning Algorithms. Using this, we can predict the possibility of storms, rain, snow, tornadoes, typhoons, tsunamis, and many more natural calamities. Nested domain simulations are weather simulations that include subtask simulations and several regions of interest[3]. The nested domains are typically simulated at a higher spatio-temporal resolution, i.e., grid points are closer together, and more integration steps (simulations) are performed for the nested domains compared to the parent domain[4]. Also, Meteorologists are researching different algorithms to predict more realistic situations.

Here further, we will discuss the evaluation of weather forecasts and how it is done.

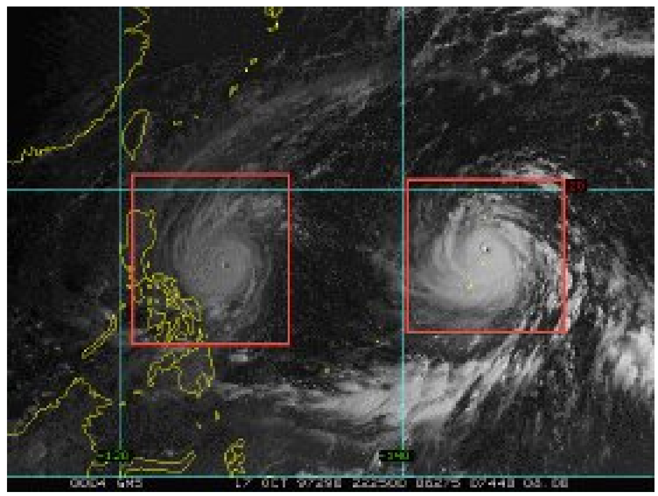


Fig. 1. Image of Storm Creation [5]

In the above figure, we can see that it is an image of a storm creating. This is a single image. Such images in large numbers are clubbed together and fed to the deep learning model. Thus, the deep learning model prepared with the help of the algorithm will train itself with the help of the provided data. Then the learned model will recognize the storms that will be created or the existence of storms. This is how the evaluation of weather can be done with the help of deep Learning. A storm is just an example. Likewise, many other events can be evaluated using such deep learning models.

The above approach is called Prediction using image classification, where images are the input, and the model learns with the help of the mages. In other events, numerical data can also be used to learn from the model. However, there is Prediction, but it is only a few percent accurate most of the time. Such models can be trained multiple times to increase their accuracy. This is the way how we can evaluate events using deep learning algorithms. Further, we will explore evaluation and its limitations, what deep Learning is, and the implementation of weather evaluation using simple Recurrent Neural Network(RNN) which is a deep learning algorithm.

II. Weather Evaluation

As time goes by, we see the concept of climate change getting real. Weather evaluations over the years have also resonated the same. We need to devise a plan to combat climate change, or else the demand and supply of food production and liveability will be in danger. For this, it has become necessary to predict the weather with current parameters to prepare ourselves better to combat climate change. To do this, we must evaluate the weather and predict future weather to prepare accordingly.

This is one of many reasons for weather evaluation. If we can predict and evaluate weather, we can also make decisions relating to agriculture. Agriculture depends on the weather, and so does the output from agriculture. Better weather prediction, better agricultural produce. We can manage the production output by predicting the weather. Similarly, there are many reasons for evaluating weather, and it becomes essential.

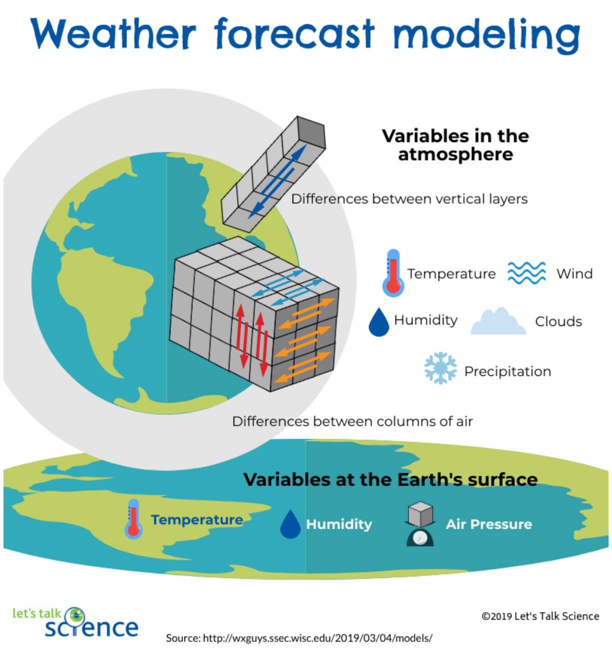


Fig. 2. Weather Forecast Modeling [6]

There exist different vertical layers above the earth's surface. This results in different variables at different vertical layers. These variables are Temperature, wind, Humidity, Clouds, and precipitation. All these variables depend on various factors at a given point in time. These predictions do have limitations. Even after using learned models, predicting the weather over the next few days is challenging. And this is mainly because of three things,Available data, time available, and weather complexity. Also, weather can be unpredictable; sometimes, slight changes in weather change the entire weather of that area t a specific time. Weather predictions can be biased as most weather observation stations are in cities. Due to this, rural areas are either neglected or have a small amount of data, which is not enough for predictions. This gives out partial data for cities but is also applied to nearby rural areas, which turns out to be wrong predictions. Although, in such cases, scientists use satellite data to cover up the margins; it reduces the model's overall accuracy. This happens because of the cloud cover and moisture content present in the air. The weather model is made up of mathematics and software to train itself and provide an accurate output. Another limitation is the resolution of the grid. Typically low resolution is not an issue. However, low resolution might result in significant mistakes in areas where surface characteristics change significantly over short distances[7].

Moreover, it is challenging to predict different atmospheric variables like Temperature, Wind, Humidity, Clouds, and Precipitation. Different regions have different weather conditions, and using the same parameters for different regions would result in inaccurate predictions. That is why weather evaluations are compassionate and slight changes matter. Extreme temperatures can occur in Europe, for example, when the atmosphere is blocked, but the local processes that cause summer and winter temperature extremes are different (advection of cold air)[8].

Weather models are still getting better. Research on such models is making the models better and also increasing the speed of computation. However, despite the expansion of observational networks and weather models' complexity, meteorological forecasting errors will always occur[9]. Such errors will only result in more research being done in numerical weather detection models or weather detection with image classification, with a chance to increase the number of factors or parameters and the accuracy of the weather models.

III. Deep Learning

According to [10], "deep learning is a class of machine learning algorithms that: (1) use a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input, (2) learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts."

Research psychologist Gary Marcus noted [11]: "Realistically, deep learning is only part of the larger challenge of building intelligent machines. Such techniques lack ways of representing causal relationship (. . .) have no obvious ways of performing logical inferences, and they are also still a long way from integrating abstract knowledge, such as information about what objects are, what they are for, and how they are typically used. The most powerful AI systems, like Watson (. . .) use techniques like deep learning as just one element in a very complicated ensemble of techniques, ranging from the statistical technique of Bayesian inference to deductive reasoning."

Two steps are involved in training a deep neural network for classification, and they are described:
1. Unsupervised SdA training
2. Initialize a multilayer neural network with the trained SdA's weights and train it using gradient descent. [12]

Supervised learning is a machine learning approach[21] that uses labeled datasets. labeled datasets increases the accuracy of the model and can be trained with time. Supervised learning or training can be divided into two types namely Classification and Regression.

Unsupervised learning is a machine learning approach that works with unlabeled datasets. The machine detects patterns through the data on its own. Unsupervised Learning or training can be divided into three types namely Clustering, Association and Dimensionality reduction.

Both these approaches have their own strengths and it depends on the type and volume of the data to figure out which one to use. There is also third type of training called as semi-supervised learning where the model uses labeled datasets and work on the similar lines of unsupervised learning or training.This is mostly used in medical field.

A. Neural Network

A neural network is a type of data processing system whose architecture is modeled after the cerebral cortex region of the brain. It consists of numerous simple, highly interconnected processing components[13]. Neural network consists of input layer, output layer, and has weights and threshold and some hidden layers. All these layers are connected through neurons.

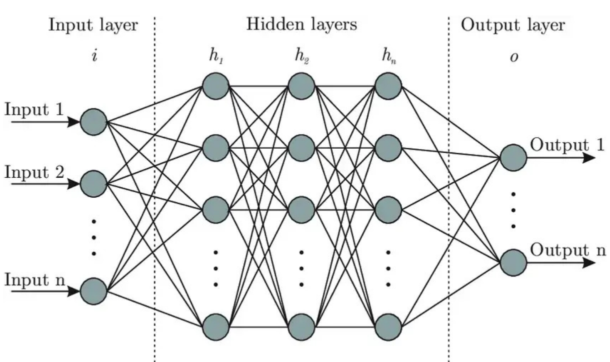


Fig. 3. Neural Network Structure [14]

As soon as the value of a neuron surpasses the specified threshold of that neuron its get activated at the next layer. Similarly the neuron will get activated at every layer in between till it reaches output layer. The structure of the neural network resembles the network of neurons inside of human brain. As you can see n fig.3, there's a input layer, output layer and hidden layers. As soon as the neurons in the input layer i gets activated, it is passed to the hidden layers. In the fig3, the neurons which are activated in input layer i are passed on the next hidden layer h1. Similarly, the activated neurons get passed onto hidden layers h2 till h3 and then to the output layer. This will display the result. There are lot of Deep learning models. The common models are Autoencoder (AE), Deep Belief Network (DBN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN)[15]. We focus on Recurrent Neural Network(RNN) in this section.

B. Recurrent Neural Network (RNN)

According to the [16], "Recurrent Neural Network is a kind of artificial neural network. Apart from having the structure of the feedforward neural network, there exists directed cycles in RNN. This structure allows the information to be circulated in the network, so the output of each time is not only related to the input at present, but related to the input at previous timestamps."

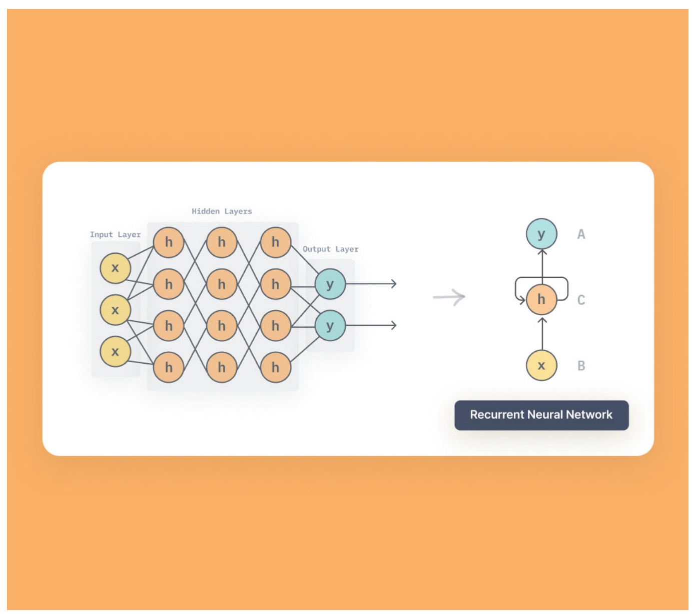
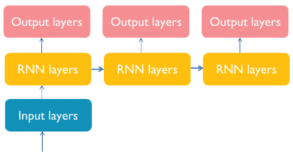


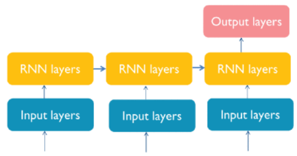
Fig. 4. Recurrent Neural Network [17]

In RNN's we add a loop to the traditional network which allows passing of information from one step to next. This information is in hidden states. Instead of multiple hidden layers, we can have just one hidden layer and add a loop to that one hidden layer so that it goes through that loops as many times as we want. It is mostly used for sequential data and time-series algorithm. Its unique feature as an algorithm is that it has short-term memory, which enables RNN to has the ability to go back to retrieve previous information[18]. There are four types of Recurrent neural networks :
1. One to many[19]

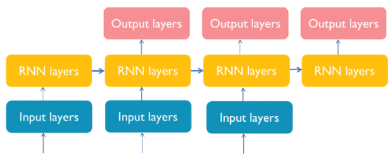
2. Many to one[19]
3. Many to many[19]
4. One to one[19]



(a) One to Many RNN.



(b) Many to One RNN.



(c) Many to Many RNN.

Fig. 5. RNN time steps [20]

IV. Use Case

Here, In this section, we will see the implementation of a weather evaluation model using deep Learning. As this is Sequential data, we will use Recurrent Neural Network (RNN), an algorithm for deep Learning. The dataset consists of weather metrics like Temperature, Pressure, and Humidity for cities worldwide, focusing primarily on North American cities. The dataset is for the period from 2012- 2017. There are over 45000 data points recorded across the dataset. We will be making time-series predictions for weather evaluation. For this, we will be using past weather data for predictions. For this, we will focus on the data from the city of San Francisco. This dataset can be found on Kaggle by the name "Historical Hourly Weather Data 2012-2017".

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
pd.set_option('mode.chained_assignment', None)
from keras.models import Sequential
from keras.layers import Dense, SimpleRNN
from keras.optimizers import RMSprop
from keras.callbacks import Callback
.
.
.
.
humidity_Sanfro = humidity[['datetime','San Francisco']]
temp_Sanfro = temp[['datetime','San Francisco']]
pressure_Sanfro = pressure[['datetime','San Francisco']]
```

We started with Data preprocessing. In data preprocessing, we started importing the required libraries like pandas, numpy, matplotlib, and keras. Then we imported the data using pandas function pd.read. Since we are focusing on the city of San Francisco, we made new dataframes in which extracted san francisco data is included. The dataset includes Temperature, Humidity, and Pressure for the city of San Francisco.

After taking a closer look at the data, we can see many NaN values in the dataset. After processing, we concluded that there are 942 NaN values for Humidity and 793 NaN values for Temperature, and 815 NaN values for Pressure. Moving further, we choose 10000 training points, which means we will train 10000 points, and the model will predict the following 35000 points.

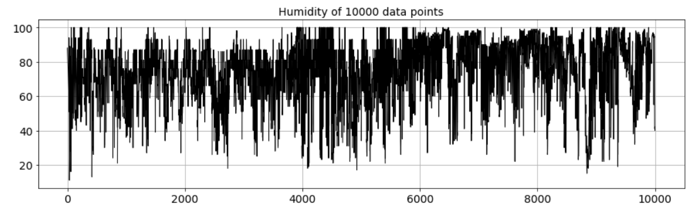


Fig. 6. Humidity of 10k points

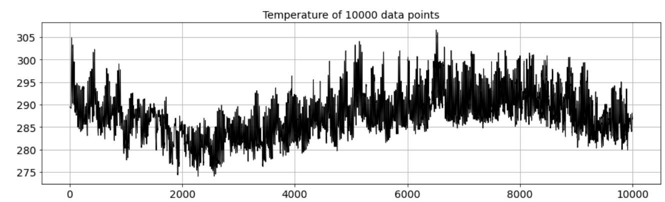


Fig. 7. temperature of 10k points

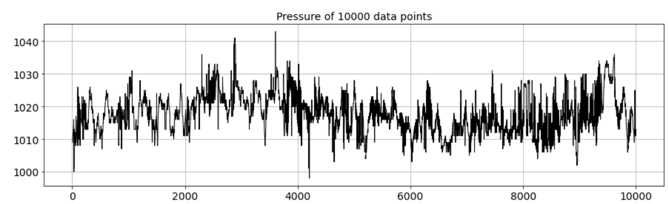


Fig. 8. pressure of 10k points

These are the plots for the 10k data points for Humidity, Temperature, and Pressure. Then we fill up NaN values by using interpolation. Further, we split the data into train and test data. For this, we split the data into trains until the first 10000 data points, and the rest 35000 are included in the test dataset. Then we reshape the train and test dataset to a 2D array with dimensions -1 to 1. We can also plot the test and train data together in a single plot. Here we use step=10. In Recurrent Neural Network(RNN), whatever the number of elements is there as output or the y prediction will be a step of that number. For example, for step=1, the y prediction will be 2. This is also called embedding size. In short, from 1 step to 10 steps of data can predict the data for step 11. The command .shape then gives us the train and test data size. Train data is 10010 points, and test data is 35262 points.

Now we will convert test and train data into a multi dimensional array as 3D matrix. For this we again the reshape the data as before with dimensions as 0,1,1. Now the shape of training and test set becomes 10000 for training set and 35252 for test data.

```
def build_simple_rnn(num_units=120, embedding=10, num_dense=32,
lr=0.001):

    model = Sequential()
    model.add(SimpleRNN(units=num_units,
    input_shape=(1,embedding), activation="relu"))
    model.add(Dense(num_dense, activation="relu"))
    model.add(Dense(1))
    model.compile(loss=’mean_squared_error’,
    optimizer=RMSprop(lr=lr),metrics=[’mse’])

    return model
model_humidity = build_simple_rnn(num_units=120,num_dense=32,
embedding=10,lr=0.0005)
model_humidity.summary()
class MyCallback(Callback):
    def on_epoch_end(self, epoch, logs=None):
        if (epoch+1) % 50 == 0 and epoch>0:
            print("Epoch number {} done".format(epoch+1))
batch_size=10
num_epochs = 1000
model_humidity.fit(trainX,trainY,
          epochs=num_epochs,
          batch_size=batch_size,
          callbacks=[MyCallback()],verbose=0)
```

Here we build the simple Recurrent Neural Network(RNN) model for the humidity data set. We will use Relu as an Activation function. This model includes one input layer, one output layer, and one hidden layer with a loop. Used RMSprop as optimizer and loss function as mean squared error. Used steps per epoch as 120 with 1000 epochs to run. Steps per epoch are calculated as the number of training samples divided by the batch size. Here we used a learning rate of 0.005. Here we use the Callback function to print the update of epochs after every 50 epochs. In Fig.9, we can see the true and predicted data for Humidity.

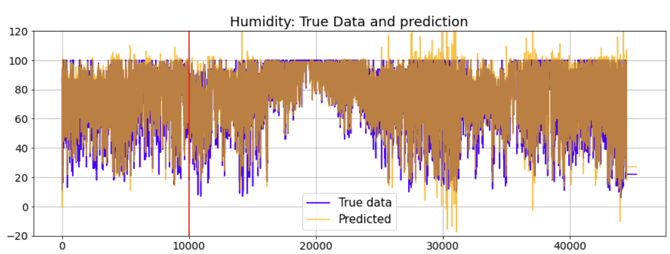


Fig. 9. Prediction for Humidity

Now we will start with the modeling of the temperature data. Similarly like Humidity data we will do it for temperature data. We will start with reshaping the training and test data of temperature dataset into 2D array with dimensions -1 and 1. Then we convert it into a multi dimensional array like a 3D matrix with dimensions 0,1,1. Here we again build a simple Recurrent model(RNN) for temperature data. We will use the same activation function, loss function and optimizer as Humidity data model. Used steps per epoch as 128 with 2000 epochs to run. We are using batch size as 8 and learning rate of 0.005. In Fig.10, we can see the true data and the predicted data for temperature.

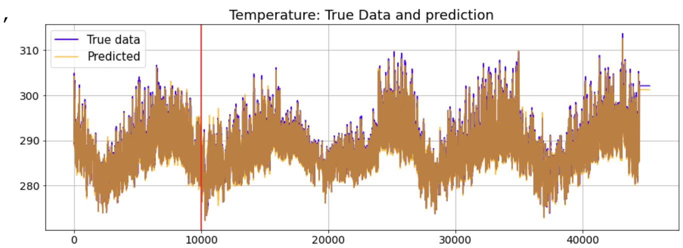


Fig. 10. Prediction for Temperature

Here we will start with the modeling of the Pressure data.Similarly like Humidity and temperature data we will do it for pressure data. We will start with reshaping the training and test data of pressure dataset into 2D array with dimensions -1 and 1. Then we convert it into a multi dimensional array like a 3D matrix with dimensions 0,1,1. Here we again build a simple Recurrent model(RNN) for pressure data. We will use the same activation function, loss function and optimizer as of Humidity data model and temperature data model. Used steps per epoch as 128 with 500 epochs to run. We are using batch size as 8 and learning rate of 0.005. In Fig.11, we can see the true data and the predicted data for Pressure.

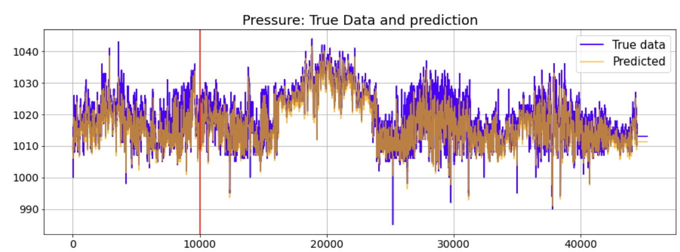


Fig. 11. Prediction for Pressure

It is evident that with just 10000 datapoints it is able to predict the general shape pretty accurately.

V. Future Work

In the above use case we can see that the model works pretty fine. This can be further tuned with respect to more training points so as the machine can be trained with more accuracy. Computational limitations forced us to work with less data points. This model can be further tuned and used for similar larger datasets for weather evaluation around the world. Currently, this model works on three features humidity, temperature and pressure. But In future this number of features can be increased. Although with more features it will require more tuning and more computational capabilities but this will prove to be more accurate.

VI. CONCLUSION

Deep learning for Weather evaluation is the best possible approach available right now and needs to be more researched. More deep learning models need to be designed for weather evaluation as it will prove very important for the entire world as we can already feel the effects of climate change and also agricultural produce can be increased which would result in better economy around the world. Though the impact of weather evaluation looks minimal, is not the actual case. If weather evaluation is not doe, it has an direct impact on economy and changes people's livelihood. In this paper, it can be seen the importance of weather evaluation along with its limitations and also how deep learning algorithm helps in weather evaluation.

REFERENCES

[1] G. S. Gill, V. Saxena, R. Mittal, T. George, Y. Sabharwal and L. Dagar, "Evaluation and enhancement of weather application performance on Blue Gene/Q," 20th Annual International Conference on High Performance Computing, 2013, pp. 324-332, doi: 10.1109/HiPC.2013.6799138.

[2] Y. Gao, H. Ma and Y. Wang, "Research on Analysis and Evaluation of Extreme Weather Based on Dynamic Evaluation Model," 2022 International Conference on Big Data, Information and Computer Network (BDICN), 2022, pp. 196-201, doi: 10.1109/BDICN55575.2022.00045.

[3] G. S. Gill, V. Saxena, R. Mittal, T. George, Y. Sabharwal and L. Dagar, "Evaluation and enhancement of weather application performance on Blue Gene/Q," 20th Annual International Conference on High Performance Computing, 2013, pp. 324-332, doi: 10.1109/HiPC.2013.6799138.

[4] G. S. Gill, V. Saxena, R. Mittal, T. George, Y. Sabharwal and L. Dagar, "Evaluation and enhancement of weather application performance on Blue Gene/Q," 20th Annual International Conference on High Performance Computing, 2013, pp. 324-332, doi: 10.1109/HiPC.2013.6799138.

[5] G. S. Gill, V. Saxena, R. Mittal, T. George, Y. Sabharwal and L. Dagar, "Evaluation and enhancement of weather application performance on Blue Gene/Q," 20th Annual International Conference on High Performance Computing, 2013, pp. 324-332, doi: 10.1109/HiPC.2013.6799138.

[6] https://letstalkscience.ca/educational-resources/stem-in-context/why-weather-so-hard-predict

[7] https://letstalkscience.ca/educational-resources/stem-in-context/why-weather-so-hard-predict

[8] Sillmann, Jana Thorarinsdottir, Thordis Keenlyside, Noel Schaller, Nathalie Alexander, Lisa Hegerl, Gabriele Seneviratne, Sonia Vautard, Robert Zhang, Xuebin Zwiers, Francis. (2017). Understanding, modeling and predicting weather and climate extremes: Challenges and opportunities. Weather and Climate Extremes. 18. 10.1016/j.wace.2017.10.003.

[9] https://letstalkscience.ca/educational-resources/stem-in-context/why-weather-so-hard-predict

[10] L. Deng and D. Yu, "Deep Learning: Methods and Applications", Foundations and Trends in Signal Processing, vol. 7, no. 3–4, pp. 197-387, 2014.

[11] Gary Marcus, "Is "Deep Learning" a Revolution in Artificial Intelligence?" in The New Yorker, 2012.

[12] F. Q. Lauzon, "An introduction to deep learning," 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), 2012, pp. 1438-1439, doi: 10.1109/ISSPA.2012.6310529.

[13] R. E. Uhrig, "Introduction to artificial neural networks," Proceedings of IECON '95 - 21st Annual Conference on IEEE Industrial Electronics, 1995, pp. 33-37 vol.1, doi: 10.1109/IECON.1995.483329.

[14] https://towardsdatascience.com/designing-your-neural-networks-a5e4617027ed

[15] X. Du, Y. Cai, S. Wang and L. Zhang, "Overview of deep learning," 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), 2016, pp. 159-164, doi: 10.1109/YAC.2016.7804882.

[16] X. Du, Y. Cai, S. Wang and L. Zhang, "Overview of deep learning," 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), 2016, pp. 159-164, doi: 10.1109/YAC.2016.7804882.

[17] https://www.v7labs.com/blog/recurrent-neural-networks-guide

[18] Y. Chen and J. Li, "Recurrent Neural Networks algorithms and applications," 2021 2nd International Conference on Big Data Artificial Intelligence Software Engineering (ICBASE), 2021, pp. 38-43, doi: 10.1109/ICBASE53849.2021.00015.

[19] N. M. Rezk, M. Purnaprajna, T. Nordström and Z. Ul-Abdin, "Recurrent Neural Networks: An Embedded Computing Perspective," in IEEE Access, vol. 8, pp. 57967-57996, 2020, doi: 10.1109/ACCESS.2020.2982416.

[20] N. M. Rezk, M. Purnaprajna, T. Nordström and Z. Ul-Abdin, "Recurrent Neural Networks: An Embedded Computing Perspective," in IEEE Access, vol. 8, pp. 57967-57996, 2020, doi: 10.1109/ACCESS.2020.2982416.

[21] Supervised Learning - A Complete Introduction Wiki - Explorium. https://www.explorium.ai/wiki/supervised-learning/

VII. DECLARATION OF ORIGINALITY

I, Pritish Samant, herewith declare that I have composed the present paper and work by myself and without use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form has not been submitted to any examination body and has not been published. This paper was not yet, even in part, used in another examination or as a course performance.