ISSN: 0974-5823 Vol. 7 No. 2 February, 2022

*International Journal of Mechanical Engineering*

# Light-Weight Real Time Weather Forecasting Simulation over Bangladesh using Deep Learning

**Lingala Thirupathi[1], C Kishor Kumar Reddy[2], Anisha P R[3], D Rambabu[4] and Rajashekar Parupati[5]**

[1,2,3] CSE Dept, Stanley College of Engineering and Technology for Women, Telangana, India.

[4]Asst. Prof, CSE Dept, Sreendhi Institute of Science & Technology, Telangana, India.

[5]Asst. Prof, CSE Dept, Vidya Jyothi Institute of Technology, Telangana, India

**Abstract**

The question of predicting weather has baffled mankind for centuries but with machine learning techniques it is now possible to predict weather with good accuracy. This chapter proposes to predict weather more accurately within the context of Bangladesh. As predicting weather is a challenging task, the application of machine learning in this sector has promising results. The people of Bangladesh suffer a lot due to bad weather pattern and it is an on-going problem. This study hopes to achieve an insight on how machine learning techniques can prove to be helpful using classification algorithms to predict cloud patterns based on past data of Dhaka. Using state of the art classifiers from statistical models such as Naïve Bayes, Decision Tree, Logistic Regression and non-probabilistic models like Support Vector Machine, gives accurate results up to 70% and more when predicting weather patterns.

**Keywords:** Weather, Forecasting, Machine Learning, Deep Learning, Logistic Regression, Support Vector Machine, Naïve Bayes, Multinomial, Decision Tree.

## 1. INTRODUCTION

The recent climate is quite unpredictable in the context of Bangladesh and as a result a lot people are faced with destruction and calamity. This affects a large population of farmers that are faced with tough decisions as crops are destroyed at the onset of bad weather. Due to unpredictable patterns, most outdoor activities have to consider the effects of weather before an event. People are faced with the difficult task of planning ahead and plans have to be changed at the last moment if an unforeseen change of events occur such as a heavy downpour. This inconvenience forces alternate measures and costs people valuable time and resources [16].

Bangladesh is also very prone to floods and rain is a constant threat to the infrastructure of cities like Dhaka which are unplanned and densely populated. Weather data from the past is an invaluable source for finding patterns and co-relations among certain weather variables [17]. The weather data is constantly monitored by government stations installed at certain parts of the country which collects various different kinds of data such as amount of rainfall, temperature, cloud, humidity. The current model for predicting weather works on the basis of Numerical Weather Prediction (NWP), which is tasked to predict future weather characteristics using present conditions. [6]

The machine learning techniques applied focus on the previous data collected from authorized government stations situated in Dhaka [18-20]. Using various implementations of already available popular Python packages like scikit-learn library to work on the dataset and prepare a model. The authenticity of the dataset is crucial in building upon the foundation of this chapter as erroneous data greatly alters outcomes of the algorithms. The efficiency of the algorithm greatly depends on a good dataset that is varied and not heavily skewed. The assumption that the data available also includes the possibility of anomalies due to faulty measurements and bad records. The advent of such conditions has to be dealt with in such a manner that does not disrupt the structure of the dataset so that the algorithms can work better even when generalizing new samples of collected data [21-23].

Over the last few decades, there has been a substantial amount of research done using numerical weather data. The data was primarily used for applying machine learning techniques. Other techniques included fuzzy logic and data mining [24-26]. The majority of weather forecasting relies on generative approaches and the underlying principals are based on numerical methods. In this chapter, we are presenting 2 case studies of weather forecasting using different methodologies.

## 2. CASE STUDY I

we have used the past three years data of climate having key attributes like rainfall, wind speed, direction, temperature, etc and predict calamity on the basis of these key attributes for next year three years using logistic regression technique and also perform descriptive data analysis [27].

### 2.1 METHODOLOGY

We have used Python 3.6.15, packaged by conda-forge and Jupyter notebook server 6.3.0 to predict the weather.

Importing all the necessary packages like, os, numoy, pandas, matplotlib.pyplot, seaborn, warnings. The total number of columns (23) in this dataset are shown in Table 1, column wise unique values are also shown in Table.2

Table 1: Dataset Columns Description

| Sno | Column | Non Null | Count | Dtype |
|---|---|---|---|---|
| 0 | Date | 145460 | non-null | object |
| 1 | Location | 145460 | non-null | object |
| 2 | MinTemp | 143975 | non-null | float64 |
| 3 | MaxTemp | 144199 | non-null | float64 |
| 4 | Rainfall | 142199 | non-null | float64 |
| 5 | Evaporation | 82670 | non-null | float64 |
| 6 | Sunshine | 75625 | non-null | float64 |
| 7 | WindGustDir | 135134 | non-null | object |
| 8 | WindGustSpeed | 135197 | non-null | float64 |
| 9 | WindDir9am | 134894 | non-null | object |
| 10 | WindDir3pm | 141232 | non-null | object |
| 11 | WindSpeed9am | 143693 | non-null | float64 |
| 12 | WindSpeed3pm | 142398 | non-null | float64 |
| 13 | Humidity9am | 142806 | non-null | float64 |
| 14 | Humidity3pm | 140953 | non-null | float64 |
| 15 | Pressure9am | 130395 | non-null | float64 |
| 16 | Pressure3pm | 130432 | non-null | float64 |
| 17 | Cloud9am | 89572 | non-null | float64 |
| 18 | Cloud3pm | 86102 | non-null | float64 |
| 19 | Temp9am | 143693 | non-null | float64 |
| 20 | Temp3pm | 141851 | non-null | float64 |
| 21 | RainToday | 142199 | non-null | object |
| 22 | RainTomorrow | 142193 | non-null | object |

**Table 2:** Column wise unique values Description

| Column Name | Unique Values |
|---|---|
| Date | 3436 |
| Location | 49 |
| MinTemp | 390 |
| MaxTemp | 506 |
| Rainfall | 682 |
| Evaporation | 359 |
| Sunshine | 146 |
| WindGustDir | 17 |
| WindGustSpeed | 68 |
| WindDir9am | 17 |
| WindDir3pm | 17 |
| WindSpeed9am | 44 |
| WindSpeed3pm | 45 |
| Humidity9am | 102 |
| Humidity3pm | 102 |
| Pressure9am | 547 |
| Pressure3pm | 550 |
| Cloud9am | 11 |
| Cloud3pm | 11 |
| Temp9am | 442 |
| Temp3pm | 503 |
| RainToday | 3 |
| RainTomorrow | 3 |
| RainToday | 3 |
| RainTomorrow | 3 |

Interpolation technique is applied for *filling missing values by using:*

```
data['MinTemp']= data['MinTemp'].interpolate(method='nearest')
data['MaxTemp']= data['MaxTemp'].interpolate(method='nearest')
```

*The Minimum and Maximum Temperature Comparison based on same location by using:*

```
ax = data[["Location","MinTemp", "MaxTemp"]].plot(x='Location',kind='line',color=["g","b"],rot=45)
ax.legend(["MinTemp", "MaxTemp"])
```

The comparison of temperatures based on location is depicted in Figure 1.

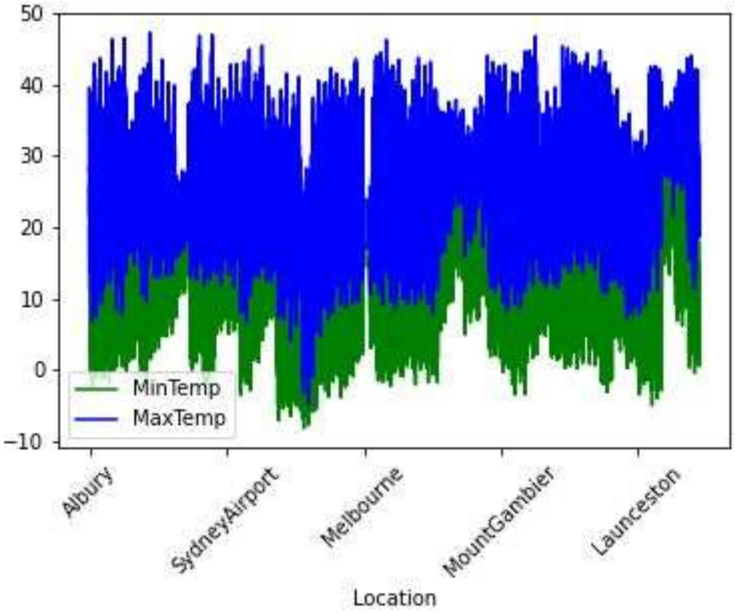


**Fig.1** Location Vs Temp

*We Took mean temperature from both different columns and stored into new one as shown in Fig.2 as below.*

```
data['Temp'] = data[['MinTemp', 'MaxTemp']].mean(axis=1)
data.plot(x='Location', y='Temp')
```

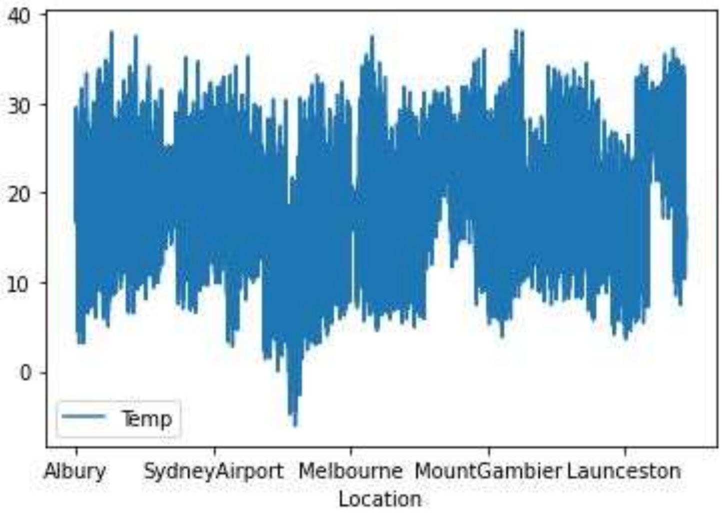


**Fig. 2** Location Vs Mean temperature.

*The most min temp is calculated as shown below and represented in table 3.The obtained* minimum temperature is -8.2 and same is represented in Table 3.

```
print("Minimum Temperature   "+str(data['MinTemp'].min()))
data.groupby('Location',sort = False)['MinTemp'].min().nsmallest()
```

**Table 3**: Calculated Minimum Temperatures

| Location | Min Temperature |
|---|---:|
| MountGinini | -8.2 |
| Canberra | -7 |
| Tuggeranong | -6.5 |
| Ballarat | -5.6 |
| AliceSprings | -5 |

*The Most max temp is calculated as shown below and represented in Table 4.The obtained* maximum temperature is 47.3.

```
print("Maximum Temperature   "+str(data['MaxTemp'].max()))
data.groupby('Location',sort = False)['MaxTemp'].max().nlargest()
```

**Table 4**: Calculated Maximum Temperatures

| Location | Max Temperature |
|---|---:|
| Moree | 47.3 |
| Richmond | 47 |
| Penrith | 46.9 |
| Woomera | 46.8 |
| Cobar | 46.6 |
| Penrith | 46.9 |
| Woomera | 46.8 |
| Cobar | 46.6 |

The Largest amount of rainfall is calculated as shown below and represented in Table 5.The obtained Highest Rainfall is 247.2

```
print("Highest Rainfall     "+str(data['Rainfall'].max()))
data.groupby('Location',sort = False)['Rainfall'].max().nlargest()
```

**Table 5**: Calculated Highest Rainfall

| Locations | Highest Rainfall |
|---|---:|
| Cairns | 247.2 |
| Newcastle | 240 |
| Williamtown | 225 |
| CoffsHarbour | 219.6 |
| Darwin | 210.6 |

The missing values mean is computed using data.isnull().mean() and is shown in Table 6.

**Table 6**: Computed missing values mean

| **Column Name** | **Missing values Mean** |
|---|---:|
| Year | 0 |
| Date | 0 |
| Location | 0 |
| MinTemp | 0 |
| MaxTemp | 0.000023 |
| Rainfall | 0.027677 |
| Evaporation | 0.54687 |
| Sunshine | 0.626142 |
| WindGustDir | 0.059876 |
| WindGustSpeed | 0.059403 |
| WindDir9am | 0.063476 |
| WindDir3pm | 0.041875 |
| WindSpeed9am | 0.007043 |
| WindSpeed3pm | 0.036002 |
| Humidity9am | 0.023941 |
| Humidity3pm | 0.067571 |
| Pressure9am | 0.118469 |
| Pressure3pm | 0.118402 |
| Cloud9am | 0.422641 |
| Cloud3pm | 0.477656 |
| Temp9am | 0.014896 |
| Temp3pm | 0.059201 |
| RainToday | 0.027677 |
| RainTomorrow | 0.027834 |
| Temp | 0 |

Every location has different wind speed, direction, Temperature and Pressure, So we have replaced Categories features with most frequent value based on the location. Still we have missing value for WindGustDir because for few locations we have no values. We have replaced these values with the mode of complete dataset and Replaced Numerical features with mean value based on location same as Categories.

This has same problem as df_cat, we have replaced the mean value of dataset as shown below.

```
df_num['WindGustSpeed']=df_num['WindGustSpeed'].fillna(data['WindGustSpeed'].mean())

df_num['Pressure9am']=df_num['Pressure9am'].fillna(data['Pressure9am'].mean())

df_num['Pressure3pm']=df_num['Pressure3pm'].fillna(data['Pressure3pm'].mean())

d={'Yes':1,'No':0}

df_cat['RainTomorrow']=df_cat['RainTomorrow'].map(d)

df_cat['RainToday']=df_cat['RainToday'].map(d)

df_cat2=df_cat[['WindGustDir','WindDir9am','WindDir3pm','Location']]
```

Similarly, we have replaced the Categories value with the value counts as shown below.

```
df_cat2['Location']=df_cat2['Location'].map(df_cat2['Location'].value_counts())
df_cat2['WindGustDir']=df_cat2['WindGustDir'].map(df_cat2['WindGustDir'].value_counts())
df_cat2['WindDir9am']=df_cat2['WindDir9am'].map(df_cat2['WindDir9am'].value_counts())
df_cat2['WindDir3pm']=df_cat2['WindDir3pm'].map(df_cat2['WindDir3pm'].value_counts())
```

We have used the Standard Scaler to scaler as,

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(df_n)
df_scaled = pd.DataFrame(scaler.fit_transform(df_n),columns = df_n.columns)
```

We have given the following commands to plot all the graphs,

```
df_x=pd.merge(df_scaled, df_cat['RainToday'],left_index=True, right_index=True)
df_x.hist(bins=50, figsize=(20, 20))
plt.show()
```

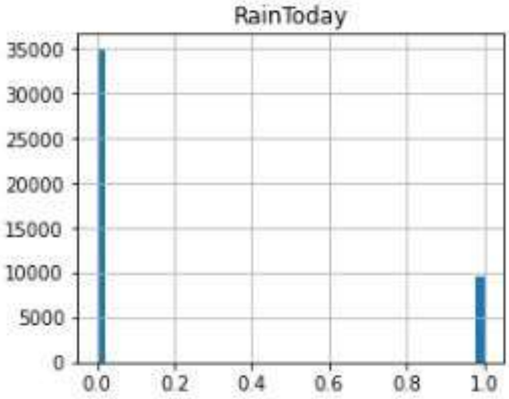


**Fig.3** RainToday graph

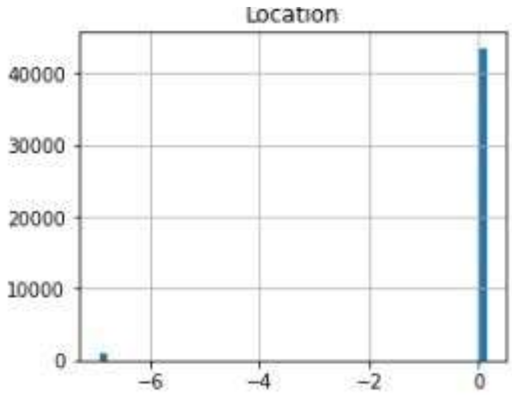


**Fig.4** Location graph

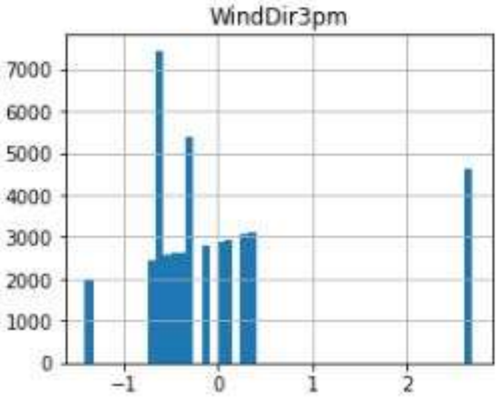


**Fig.5** WindDir3pm graph

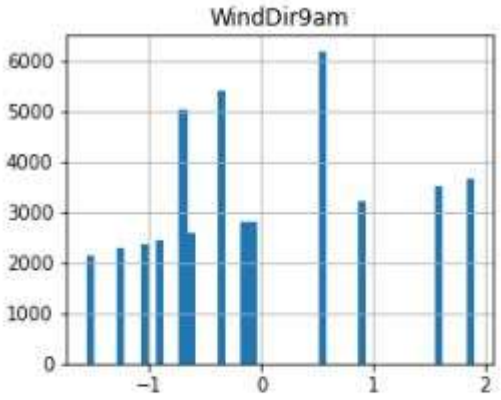


**Fig.6** WindDir9am graph

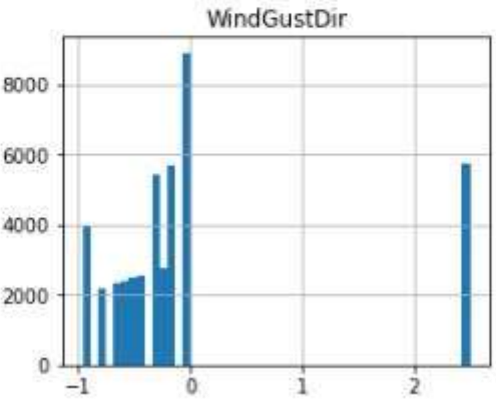


**Fig.7** WindGustDir graph

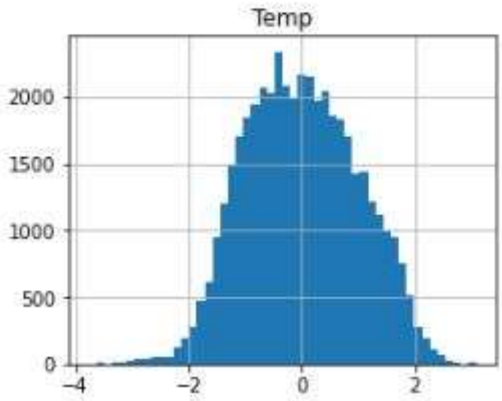


**Fig.8** Temp graph

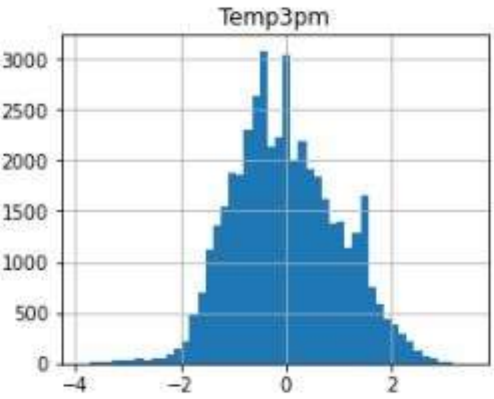


**Fig.9** Temp3pm graph

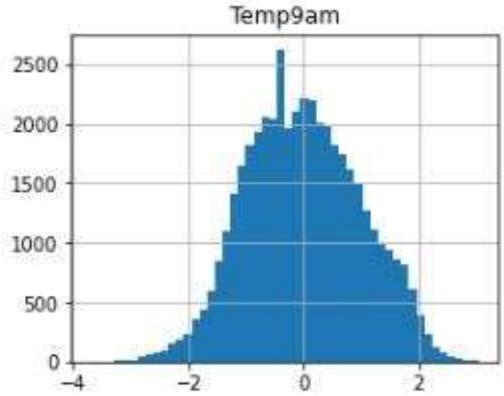


**Fig.10** Temp9am graph

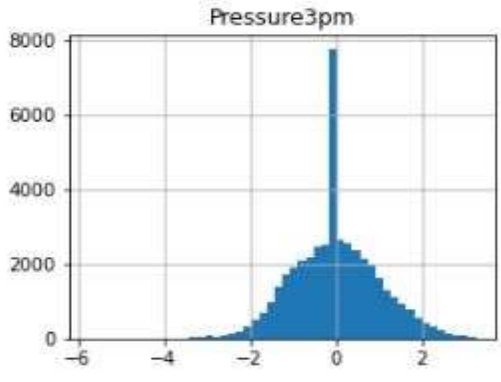


**Fig.11** Pressure3pm graph

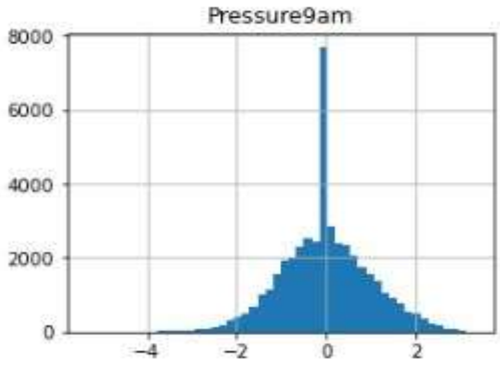


**Fig.12** Pressure9am graph

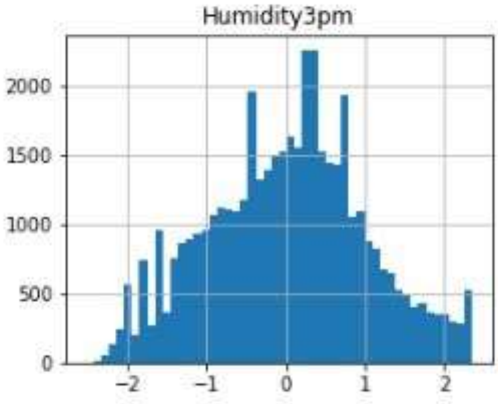
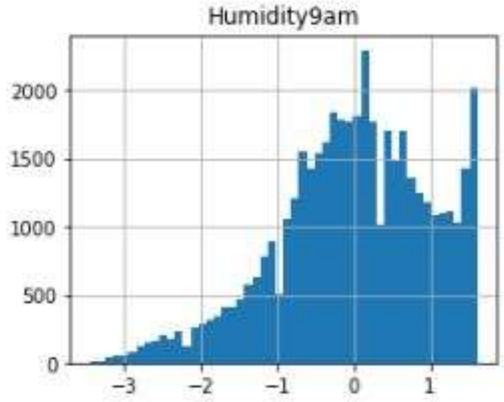


**Fig.13** Humidity3pm graph



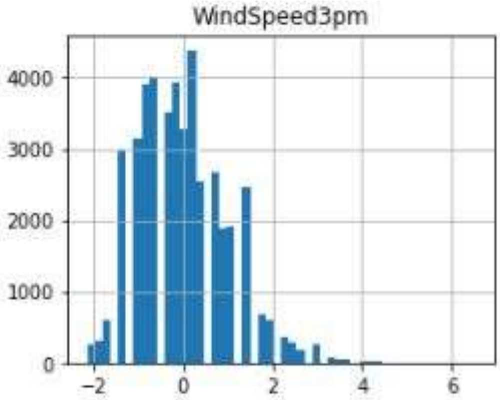
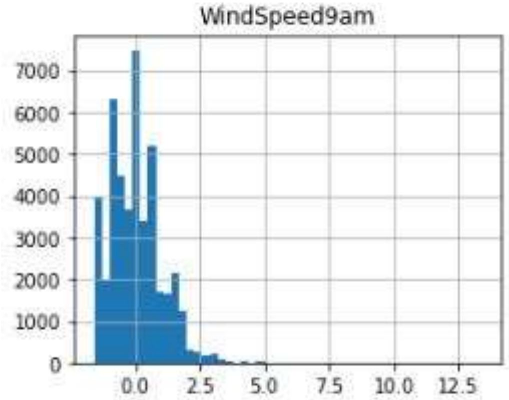


**Fig.14** Humidity 9am graph

**Fig.15** WindSpeed3pm graph



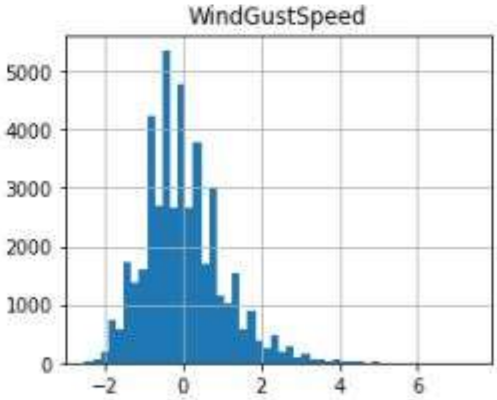
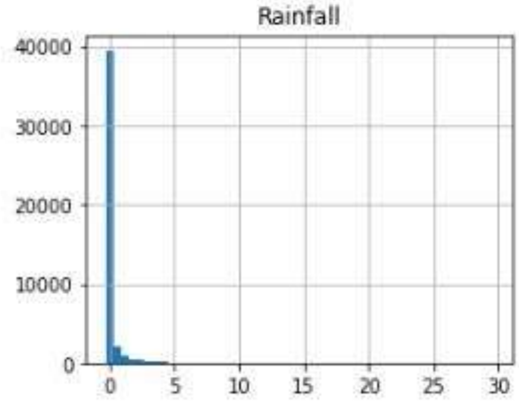


**Fig.16** WindSpeed 9am graph

**Fig.17** WindGustSpeed graph



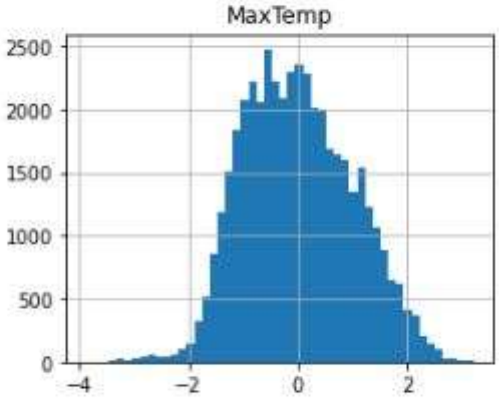


**Fig.18** Rainfall graph

**Fig.19** MaxTemp graph

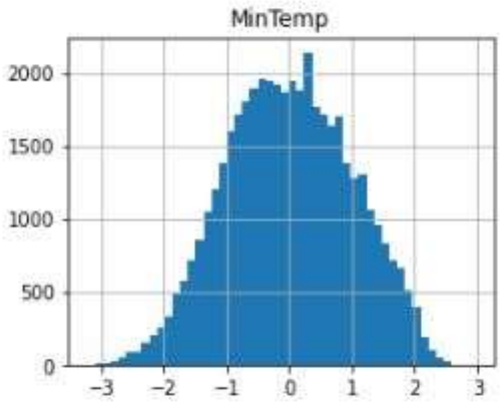


**Fig.20** MinTemp graph

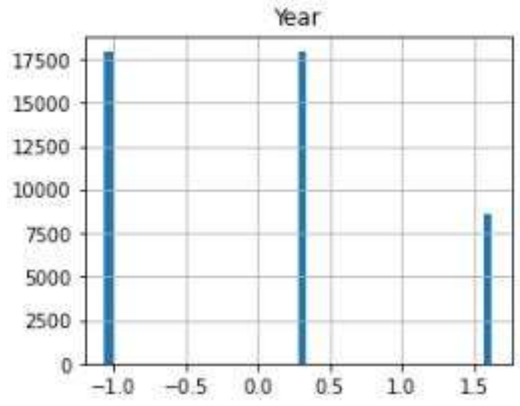


**Fig.21** Year graph

The correlation is shown below,

plt**.**figure(figsize=(20,10))

heatmap = sns**.**heatmap(df_x**.**corr(), vmin=**-**1, vmax=1, annot=**True**)

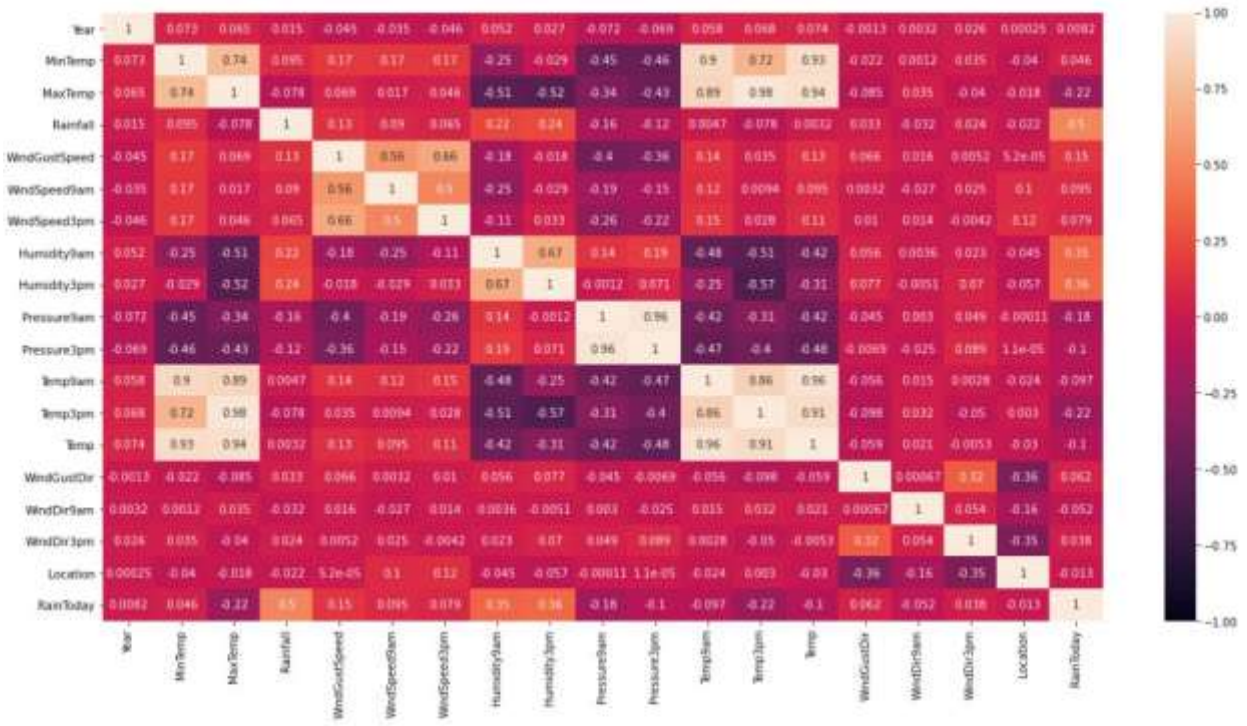


**Fig.22** The Correlation matrix

From the above matrix, we notice that the Teamp9am(89%) and Temp3pm(98%) has high correlation with MaxTemp:

df_x**.**drop('Temp9am',axis=1,inplace=**True**)

df_x**.**drop('Temp3pm',axis=1,inplace=**True**)

As we are dealing with big imbalance dataset, we have performed the oversampling as shown below.

df_cat['RainTomorrow']**.**value_counts()**.**plot(kind='barh')

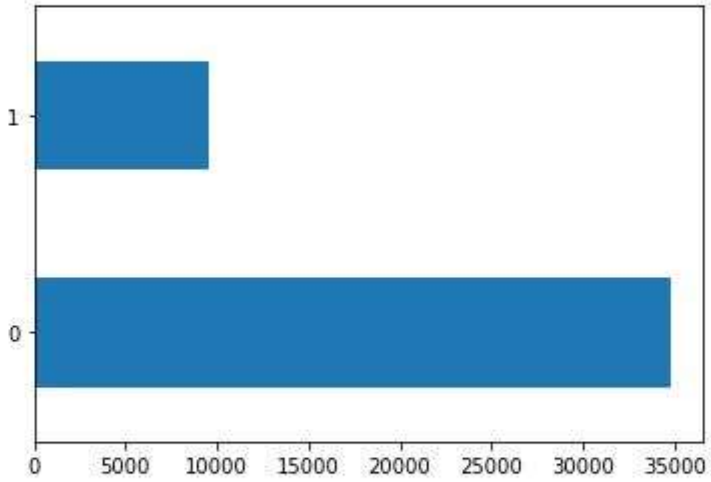


**Fig.23** Oversampling graph

The XGBoost is evaluated as,

```
from sklearn import datasets, linear_model, metrics
from sklearn.metrics import confusion_matrix, classification_report,accuracy_score
from xgboost import XGBClassifier
model = XGBClassifier(max_depth=10,random_state = 37)
model.fit(X_train_res, y_train_res)
model.score(X_train_res, y_train_res)
y_pred = model.predict(X_test)
```

The 'Confusion matrix is performed as,

```
print('Confusion matrix \n {}'.format(confusion_matrix(y_test,y_pred)))
Confusion matrix
 [[6525  453]
 [ 915  996]]
```

The 'Accuracy score  is performed as ,

```
print('Accuracy score {:.2f}'.format(accuracy_score(y_test,y_pred)*100))
Accuracy score 84.61
```

Finally we got the classification report as shown in Table 7.

```
print(classification_report(y_test,y_pred)).
```

**Table 7**: Performance measures computation

| **Parameter** | **precision** | **recall** | **f1-score** | **support** |
|---:|---:|---:|---:|---:|
| 0 | 0.88 | 0.94 | 0.91 | 6978 |
| 1 | 0.69 | 0.52 | 0.59 | 1911 |
| Accuracy | 0.72 | 0.62 | 0.85 | 8889 |
| Macro avg | 0.78 | 0.73 | 0.75 | 8889 |
| Weighted avg | 0.84 | 0.85 | 0.84 | 8889 |

The rainfall in month of December 2020 can cause flood in Australia and we can even check how wonderfully of prediction through this model is accurate that in 2020 December, Australia face this calamity [9-10].

we have used 3 key attributes like rainfall, wind and temperature and we have predicted the calamity, through proper descriptive analysis while consider different key factors with 88 precision of model our model forecast the calamity for next years from 2017 onwards with an average accuracy of 88% [28].

## 3. CASE STUDY II

Some people applied comprehensive tree-based learning algorithm. For example, N. Hasan and M.T Uddin used tree based algorithm, namely C4.5 and their output result had 96% accuracy. They also used Naïve Bayes, but the accuracy of C4.5 was much better in terms of f-score. [1] Lin and Chen [2] worked on typhoon rainfall forecasting model using ANN and their result shows that excessive spatial rainfall information may not increase the generalization of forecasting model. Awan and Awais [3] also tried to predict weather events based on fuzzy RBS method for Lahore, Pakistan. They used two different datasets of 365 examples of with only 4 features, and 2500 examples with 17 features. They mentioned their finding that fuzzy RBS method was sensitive to random sampling with replacement technique that was applied to produce. Another article reviews we found useful is S. B. Kotsiantis [4] mentioned few statistical classifiers to build classification tress. Using information entropy from a set of training examples of pre-classified samples, where each sample comprises of N-dimensional vector. H. Zhang, X. Zhao and S.Zou [5] proposed a classification algorithm naming Neuron Classification Algorithm (NCA). The algorithm has higher approximation function. They introduced the law of attraction here which increases the accuracy of weather forecast. It has been known to classify the test samples more accurately than Euclidian distance. As dataset, they chose forecast of abnormal megathermal weather in North of Zhejiang province. Combining Neural Networks and ARIMA Models for Hourly Temperature Forecast H. S. Hippert , C. E. Pedreira and R. C. Souza in their works [7] on hourly temperature forecasting, proposed to use the combination of Neural Networks and ARIMA models. The forecast is done on the basis of previous temperature records, maximum and minimum temperature data supplied by weather service. The previous temperatures are used as input to Artificial Neural Network (ANN) where there is only one output node, the predicted temperature on a particular time. Their results show that hybrid system based on ARMA model produces more accuracy than auto progressive models [29].

Simple classifiers such as Support Vector Machines (SVM) or Artificial Neural Networks (ANN) are also widely used to classify certain parameters such as rainfall and cloud states.

### 3.1 DATASET

The dataset was obtained from Bangladesh Meteoritical Department (www.bmd.gov.bd). The data set is comprised of the following variables:

Temperature(ºC): The temperature variable has great impact on precipitation and is greatly related to humidity (%). The dataset has both maximum and minimum temperatures along with the mean values.

Cloud (okta): The cloud variable measured in Okta impacts the Earth’s surface by reflecting incoming sunlight. It is also responsible for absorbing the heat emitted from surface and radiating to space. The dataset comprises of daily cloud data measured in a range from 0-8 Okta.

Wind Speed (knots): The wind variable measured in knots shows how quick the air is moving. The wind speed also has a direction and has various impacts on surface water and evaporation. The dataset comprises of daily prevailing wind speed.

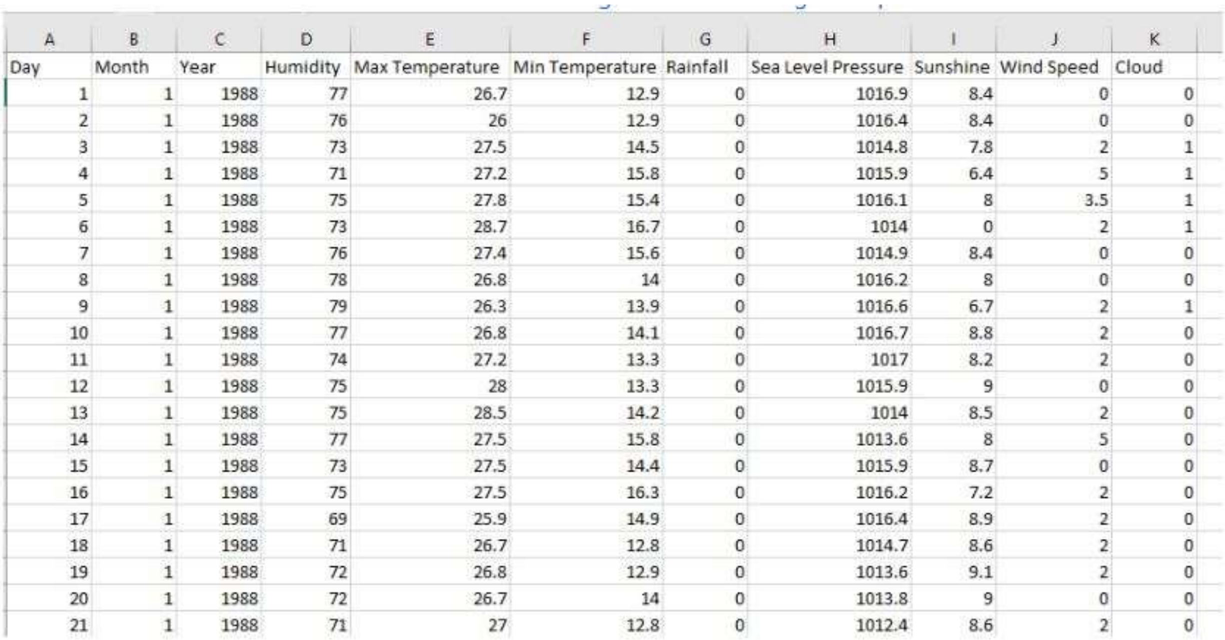
Rainfall (mm): The rainfall variable is a very important metric in weather forecasting. It helps the environment to continue to stay in its position the way it should be. Agriculture of Bangladesh mostly depends on rainfall. The dataset has daily basis of total rainfall data in millimeters.

Sunshine (hour): The sunshine variable measures the amount of sunshine at a particular place. The Sun is the basic cause of our changing weather. The day-night cycles in the weather have obvious causes and effects on weather.

Humidity (%): The humidity variable measured in percentage helps to calculate the amount of moisture in the air at a give time on a given day, which is simply the ratio of water vapor and dry air.

Sea Level Pressure (millibars): The sea level pressure variable plays a significant role in the formation of weather condition in a certain area. It is a component that has mass and weight. This means vast ocean of air inserts huge amount of pressure. So, it is natural that the air will affect the Earth’s weather.

Our dataset looks like below which we collected from Bangladesh Meteorological Department [8].



| A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|
| Day | Month | Year | Humidity | Max Temperature | Min Temperature | Rainfall | Sea Level Pressure | Sunshine | Wind Speed | Cloud |
| 1 | 1 | 1988 | 77 | 26.7 | 12.9 | 0 | 1016.9 | 8.4 | 0 | 0 |
| 2 | 1 | 1988 | 76 | 26 | 12.9 | 0 | 1016.4 | 8.4 | 0 | 0 |
| 3 | 1 | 1988 | 73 | 27.5 | 14.5 | 0 | 1014.8 | 7.8 | 2 | 1 |
| 4 | 1 | 1988 | 71 | 27.2 | 15.8 | 0 | 1015.9 | 6.4 | 5 | 1 |
| 5 | 1 | 1988 | 75 | 27.8 | 15.4 | 0 | 1016.1 | 8 | 3.5 | 1 |
| 6 | 1 | 1988 | 73 | 28.7 | 16.7 | 0 | 1014 | 0 | 2 | 1 |
| 7 | 1 | 1988 | 76 | 27.4 | 15.6 | 0 | 1014.9 | 8.4 | 0 | 0 |
| 8 | 1 | 1988 | 78 | 26.8 | 14 | 0 | 1016.2 | 8 | 0 | 0 |
| 9 | 1 | 1988 | 79 | 26.3 | 13.9 | 0 | 1016.6 | 6.7 | 2 | 1 |
| 10 | 1 | 1988 | 77 | 26.8 | 14.1 | 0 | 1016.7 | 8.8 | 2 | 0 |
| 11 | 1 | 1988 | 74 | 27.2 | 13.3 | 0 | 1017 | 8.2 | 2 | 0 |
| 12 | 1 | 1988 | 75 | 28 | 13.3 | 0 | 1015.9 | 9 | 0 | 0 |
| 13 | 1 | 1988 | 75 | 28.5 | 14.2 | 0 | 1014 | 8.5 | 2 | 0 |
| 14 | 1 | 1988 | 77 | 27.5 | 15.8 | 0 | 1013.6 | 8 | 5 | 0 |
| 15 | 1 | 1988 | 73 | 27.5 | 14.4 | 0 | 1015.9 | 8.7 | 0 | 0 |
| 16 | 1 | 1988 | 75 | 27.5 | 16.3 | 0 | 1016.2 | 7.2 | 2 | 0 |
| 17 | 1 | 1988 | 69 | 25.9 | 14.9 | 0 | 1016.4 | 8.9 | 2 | 0 |
| 18 | 1 | 1988 | 71 | 26.7 | 12.8 | 0 | 1014.7 | 8.6 | 2 | 0 |
| 19 | 1 | 1988 | 72 | 26.8 | 12.9 | 0 | 1013.6 | 9.1 | 2 | 0 |
| 20 | 1 | 1988 | 72 | 26.7 | 14 | 0 | 1013.8 | 9 | 0 | 0 |
| 21 | 1 | 1988 | 71 | 27 | 12.8 | 0 | 1012.4 | 8.6 | 2 | 0 |

**Fig 24 .** Weather Dataset Sample

We had last 30 years [1988-2017] of weather data. The training and test set is divided into two segments having 70% and 30% data split across the two categories.

**TABLE 8**: Sample Training Data

| Date | Humidity | Max Temp | Min Temp | Sunshine | Cloud |
|---|---|---|---|---|---|
| 01/01/1988 | 77 | 26.7 | 12.9 | 8.4 | 0 |
| 02/01/1988 | 76 | 26 | 12.9 | 8.4 | 0 |
| 03/01/1988 | 73 | 27.5 | 14.5 | 7.8 | 1 |
| 04/01/1988 | 71 | 27.2 | 15.8 | 6.4 | 2 |
| 05/01/1988 | 75 | 27.8 | 15.4 | 8.0 | 1 |

**TABLE 9:** Total Available Parameters

| Parameter Number | Parameter Name |
|---|---|
| 1 | Day |
| 2 | Month |
| 3 | Year |
| 4 | Humidity (%) |
| 5 | Max Temp (in $^0$C) |
| 6 | Min Temp (in $^0$C) |
| 7 | Rainfall (in mm) |
| 8 | Sea Level Pressure (in mb) |
| 9 | Sunshine (hours) |
| 10 | Wind Speed(knot) |
| 11 | Cloud (in okta) |

The dataset from BMD that was used for the chapter has past weather data from for the last 30 years (1988-2017) from Dhaka. The data collected is from a daily basis and this roughly equates to 109 samples for the entire set. The origin of the data is from Bangladesh Meteorological Department's Dhaka station with a resolution of 25 kilometers. Absolute coordinates: (Lat 23 Deg 46 Mts.N & Long 90 Deg 23 Mts.E).

The training and test set is divided into two segments having 70% and 30% data split across the two categories [30].

### 3.2 METHODOLOGY

The purpose of our chapter is to predict the cloud states of various days using other features from the dataset. The initial cloud data from the data set had a range of 0-8. This number signifies the amount of cloud on a given day, with 0 being the least cloudy and 8 being completely cloudy. The numbers in between are cloud states in the intermediate range. The cloud ranges were then compressed to reduce the original range given in the data set. This is done so that classifiers have to deal with lower number of classes when tasked with predicting. The ranges were aggregated to 0 representing the cloud states 0, 1 and 2 labelling it "clear skies". The class 1 then represented the cloud states 3, 4 and 5 giving it the label "half cloudy". The class 2 then defined the cloud states 6, 7 and 8 labelling it "fully cloudy". The cloud states are defined by the numbers ranging from 0 – 2, so this is a classification problem, namely multi-class classification. The dataset is labelled with cloud values in that range, so the learning algorithms are tasked with a supervised approach. We used several machine learning techniques which is constructed using the Python scikitlearn library. Parameter tuning of the models has been done by 5-fold cross validation using sklearn GridSearchCV.

### 3.3 MODEL SELECTION

**Multinomial Logistic Regression:** Logistic regression is known as a binary classifier but can also act as a multi-class classifier or otherwise known as multinomial logistic regression which can identify more than 2 classes using methods such as One vs All. The logistic regression calculates the probability of a class based on the hypothesis that works using the sigmoid function.

$$h_\theta(X) = \frac{1}{1 + e^{-\theta^T X}}$$

The multinomial logistic regression function is using "l1" penalization and "liblinear" with a C value of 0.01.

**Decision Tree:** Decision Tree Classifier is an algorithm suited for classification tasks and falls under the supervised criteria. Decision trees works under the basic principle that it has to predict target values and it does so by forming trees from input nodes. The decision tree used in this implementation works using the "gini" criterion [31].

$$Gini(E) = 1 - \sum cj = 1p2j$$

Here, c is the number of classes and p is the fraction of records.

**Naïve Bayes Multinomial:** Naïve Bayes Multinomial falls under the category of probabilistic classifiers. The algorithm is devised using Bayes theorem and works on the principal that there is a strong(naïve) independence between the features. Naïve Bayes classifiers are beneficial in supervised learning setup because they can be trained very efficiently. From the family of Naïve Bayes classifiers, we have implemented Gaussian Naïve Bayes, which is characterized by the following equation [32]:

$$p\left(x = v \middle| C_k\right) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

**Support Vector Machine (SVM) :** Support Vector Machines are a group of algorithms that is used for both classification and regression and are associated with supervised learning models. The most prominent feature of SVMs is that these class of algorithms fall into the "non-probabilistic" criteria. SVM works on the basis that it maximizes the Euclidean distance between the points that fall on the furthest lines and the median which is the decision boundary. A hinge loss function calculates the false classifications. When tuning for hyperparameters using the built-in GridSearchCV function using scikit-learn it is found that the algorithm works best with kernel "rbf" along with a C value of 0.01 and cv value of 5 [33].

### 3.4 RESULTS:

The results after applying the different kinds of models are discuss along with their performance measures. Models tested are: Multinomial Logistic Regression, Decision Tree, Naïve Bayes Multinomial and Support Vector Machine. The training data fed to these algorithms were 70% of the original dataset and the rest 30% for testing. The confusion matrix generated for these multi-class classifications delves a bit more on the evaluation criteria and hence are given below.

**TABLE 10**: Confusion Matrix for Logistic Regression

| | **Predicted Class** | | | |
|---|---|---|---|---|
| | | 0 | 1 | 2 |
| **Actual Class** | 0 | 1079 | 98 | 69 |
| | 1 | 148 | 389 | 343 |
| | 2 | 23 | 91 | 1048 |

**TABLE** 11: Confusion Matrix for Decision Tree

| | **Predicted Class** | | | |
|---|---|---|---|---|
| | | 0 | 1 | 2 |
| **Actual Class** | 0 | 927 | 309 | 10 |
| | 1 | 107 | 585 | 188 |
| | 2 | 14 | 217 | 931 |

**TABLE** 12: Confusion Matrix for Multinomial Naïve Bayes

| | **Predicted Class** | | | |
|---|---|---|---|---|
| | | 0 | 1 | 2 |
| **Actual Class** | 0 | 1222 | 15 | 9 |
| | 1 | 626 | 134 | 120 |
| | 2 | 370 | 361 | 431 |

**TABLE** 13: Confusion Matrix for Support Vector Machine

| | **Predicted Class** | | | |
|---|---|---|---|---|
| | | 0 | 1 | 2 |
| **Actual Class** | 0 | 1145 | 164 | 37 |
| | 1 | 126 | 493 | 261 |
| | 2 | 24 | 127 | 1011 |

The confusion matrixes of all the models are then used to find the average precision, recall and fit scores. The precision score is giving an overall general sense of the times the model is able to correctly predict and how often. The recall score is the actual relevant results correctly predicted by the model.

**TABLE** 14: Average Values of Precision, Recall and Fit-Score

| Model | Precision | Recall | F1 Score | Support |
|---|---|---|---|---|
| **Logistic Regression** | 0.76 | 0.77 | 0.76 | 3288 |
| **Decision Tree** | 0.76 | 0.74 | 0.75 | 3288 |
| **Multinomial NB** | 0.55 | 0.54 | 0.50 | 3288 |
| **SVM** | 0.77 | 0.78 | 0.77 | 3288 |

After running the models using our training set and testing it against the data that was split at the beginning for testing, it is evident that Support Vector Machines is more accurate against the test set compared to the other models. The nature of SVM allows it to run much better with a kernel trick but it becomes a lot slower when testing against the whole training set. The values of c have to be modified in order to gain better performance.

**TABLE** 15: Checking Accuracy Using Cross Validation 10

| Model | Mean Accuracy | Standard Deviation |
|---|---|---|
| **Logistic Regression** | 0.7392 | 0.010 |
| **Decision Tree** | 0.723 | 0.0219 |
| **Multinomial NB** | 0.5418 | 0.012 |
| **SVM** | 0.759 | 0.0199 |

Using cross validation value of 10, the mean accuracy and standard deviations are also collected. This shows that the results are well within range with expected errors.

**TABLE** 16: Train and Test Accuracy of the Models Tested

| Model | Training Accuracy (%) | Testing Accuracy (%) |
|---|---|---|
| **Logistic Regression** | 74.2 | 76.9 |
| **Decision Tree** | 76.8 | 74.05 |
| **Multinomial NB** | 54.27 | 54.34 |
| **SVM** | 76.42 | 77.52 |

## 4. CONCLUSIONS

After running all tests with the proposed models with the dataset provided after initial setup, the optimized parameters after tuning could predict reasonably well. One exception to this was the model Multinomial Naïve Bayes receiving just 54% in both training and test accuracy. The other models received average scores but the difference between testing and training accuracy proved to be less than 5%. This ensures that the models did not have an over fit and were performing well within ranges with errors. For future work we hope to bring in more models to test against our already optimized models and see if they generate better results. The chapter revolved around classification but there are other critical weather variables which are continuous and regression models can be used to see if they perform better or worse than their classification counterparts. A better dataset with even more variables that govern weather patterns would prove invaluable for testing out models that are better and more advanced in their prediction capabilities.

## REFERENCES

[1] N. Hasan, M.T Uddin and N.K. Chowdhury “Automated Weather Event Ananysis with Machine Learning”

[2] G.-F. Lin and L.-H. Chen, “Application of an artificial neural network to typhoon rainfall forecasting,” Hydrological Processes, vol. 19, no. 9,pp. 1825–1837, 2005.

[3] M. S. K. Awan and M. M. Awais, “Predicting weather events using fuzzy rule based system,” Applied Soft Computing, vol. 11, no. 1, pp.56–63, 2011.

[4] S.B. Kotsiantis, “Supervised machine learning: A review of classification techniques,” in Proceeding of the 2007 Conference on Emerging Artificial Intelligence Applictions in Computer Engineering: Real World AI Systems with Applications in Computer Egineering: Real World AI systems with Applications in eHealth,HCI Amsterdam, The Netherlands, The Netherlands: IOS Press, 2007, pp. 3–24.

[5] H. Zhang, X. Zhao and S.Zou, “Neuron Classification Algorithm and Magthermal Weather Forecast” Beijing:Meteorology Press, 2002.

[6] Dirmeyer, Paul A.; Schlosser, C. Adam; Brubaker, Kaye L. “Precipitation, Recycling, and Land Memory: An Integrated Analysis.”https://journals.ametsoc.org/doi/pdf/10.1175/2008JHM101 6.1 . Dec. 2016.

[7] H. S. Hippert , C. E. Pedreira and R. C. Souza, “Combining Neural Network and ARIMA models for Hourly Temperature Forcast”, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, vol 12, no. 1, pp. 57-28, 2014.

[8] http://live4.bmd.gov.bd/

[9] https://floodlist.com/australia/nsw-queensland-december-2020

[10]https://www.abc.net.au/news/2020-12-28/decade-on-from-devastating-thedore-flood-entire-town-evacuated/12985622

[11] Lingala Thirupathi and Venkata Nageswara Rao Padmanabhuni, “Multi-level Protection (Mlp) Policy Implementation using Graph Database” International Journal of Advanced Computer Science and Applications(IJACSA), 12(3), 2021. http://dx.doi.org/10.14569/IJACSA.2021.0120350.

[12] Lingala Thirupathi et al 2021 J. Phys.: Conf. Ser. 2089 012049; DOI: 10.1088/1742-6596/2089/1/012049.

[13] Thirupathi Lingala et al 2021 J. Phys.: Conf. Ser. 2089 012050; DOI: 10.1088/1742-6596/2089/1/012050.

[14] S. Pratapagiri, R. Gangula, R. G, B. Srinivasulu, B. Sowjanya and L. Thirupathi, "Early Detection of Plant Leaf Disease Using Convolutional Neural Networks," 2021 3rd Interna-tional Conference on Electronics Representation and Algorithm (ICERA), 2021, pp. 77-82, doi: 9.1109/ICERA53111.2021.9538659.

[15] Padmaja P, Sophia IJ, Hari HS, Kumar SS, Somu K, et al., (2021) Distribute the Message over the Network Using another Frequency and Timing Technique to Circumvent the Jam-mers. J Nucl Ene Sci Power Generat Techno 10:9.

[16]Lingala Thirupathi, Dr. Venkata Nageswara Rao Padmanabhuni, " protected framework to detect and mitigate attacks": International journal of analytical and experimental modal analysis, volume XII,Issue-VI,(2020) Page No: 2335-2337, DOI:18.0002.IJAEMA.2020.V12I6.200001.0156858943.

[17] Shashi Rekha, Lingala Thirupathi, Srikanth Renikunta, Rekha Gangula, Study of security issues and solutions in Internet of Things (IoT), Materials Today: Proceedings, 2021,ISSN 2214-7853, https://doi.org/10.1016/j.matpr.2021.07.295.

[18] Rekha Gangula, Lingala Thirupathi, Rajashekar Parupati, K. Sreeveda, Saritha Gattoju, Ensemble machine learning based prediction of dengue disease with performance and accuracy elevation patterns, Materials Today: Proceedings, 2021, ISSN 2214-7853, https://doi.org/10.1016/j.matpr.2021.07.270.

[19] Lingala Thirupathi and P.V. Nageswara Rao, "Developing a MultiLevel Protection Framework Using EDF", International Journal of Advanced Research in Engineering and Technology (IJARET),2020,volume:11, Issue: 10, Pages: 893-902.

[20] Lingala Thirupathi, Dr. Venkata Nageswara Rao Padmanabhuni, " protected framework to detect and mitigate attacks": International journal of analytical and experimental modal analysis, volume XII,Issue-VI,(2020) Page No: 2335-2337, DOI:18.0002.IJAEMA.2020.V12I6.200001.0156858943

[21] L. Thirupathi, G. Rekha, "Future drifts and Modern Investigation Tests in Wireless Sensor Networks" , International Journal of Advance Research in Computer Science and Management Studies, Volume 4, Issue 8 (2016).

[22] L Thirupati, R Pasha, Y Prathima, “Malwise System for Packed and Polymorphic Malware”, International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 167– 172 (2014).

[23] Thirupathi Lingala, Ashok Galipelli, Mahesh Thanneru, “Traffic Congestion Control through Vehicle-to-Vehicle and Vehicle to Infrastructure Communication”, (IJCSIT) International Journal of Computer Science and Information Technologies, volume 5, Issue 4, Pages :5081-5084 (2014).

[24] M.Swathi, L.Thirupathi, “Algorithm For Detecting Cuts In Wireless Sensor Networks” in International Journal of Computer Trends and Technology (IJCTT) – volume 4 Issue10 (2013).

[25] L Thirupathi, Y Reddemma, S Gunti - SIGCOMM Computer Communication Review, “A secure model for cloud computing based storage and retrieval”, volume 39, Issue 1, Pages: 50-55 (2009).

[26] L. Thirupathi and R.P.V. Nageswara, "Understanding the Influence of Ransomware: An Investigation on Its Development Mitigation and Avoidance Techniques", Grenze International Journal of Engineering & Technology (GIJET), vol. 4, no. 3, pp. 123-126, 2018.

[27] Sunanda Nalajala, Lingala Thirupathi, N.L.Pratap,”Improved Access Protection of Cloud Using Feedback and De-Duplication Schemes “, Journal of Xi'an University of Architecture & Technology, Volume XII, Issue IV ( 2020).

[28] V.Srividya, P.Swarnalatha, L.Thirupathi, “Practical Authentication Mechanism using PassText and OTP” in Grenze International Journal of Engineering and Technology, Special Issue,Grenze ID: 01.GIJET.4.3.27,© Grenze Scientific Society, 2018.

[29] Kishor Kumar Reddy C, Anisha P R, Shastry R, Ramana Murthy B V, “Comparative Study on Internet of Things: Enablers and Constraints”, Advances in Intelligent Systems and Computing, 2021

[30] Kishor Kumar Reddy C, Anisha P R, Apoorva K, “Early Prediction of Pneumonia using Convolutional Neural Network and X-Ray Images”, Smart Innovation, Systems and Technologies, 2021

[31] Kishor Kumar Reddy C and Vijaya Babu B, “ISPM: Improved Snow Prediction Model to Nowcast the Presence of Snow/No-Snow”, International Review on Computers and Software, 2015

[32] Kishor Kumar Reddy C, Rupa C H and Vijaya Babu B, “SLGAS: Supervised Learning using Gain Ratio as Attribute Selection Measure to Nowcast Snow/No-Snow”, International Review on Computers and Software, 2015

[33] Kishor Kumar Reddy C, Rupa C H and Vijaya Babu B, “A Pragmatic Methodology to Predict the Presence of Snow/No-Snow using Supervised Learning Methodologies”, International Journal of Applied Engineering Research, 2014.