Long term Fog prediction

Student - Manisha Chaurasia (221345)* and Supervisor - Prof. Mahendra K Verma†
Department of Physics, Indian Institute of Technology Kanpur, Uttar Pradesh 208016, India
(Dated: November 11 2023)

Fog prediction is necessary to reduce the problems of transportation and agriculture. To do this with the help of Machine learning, we collect data from the Iowa State University website. The data undergo preprocessing steps, and the output data is used to train the machine learning models. We have used three models - Base model, Model-1 and Model-2. The base model is the basic hypertunning of different machine learning models, and Model-1 and Model-2 are based on the hypothesis that the problem's output depends on its previous values. After analyzing the result, this hypothesis appears correct, and the dependent variable value depends on its last values.

INTRODUCTION

Fog is a meteorological phenomenon consisting of a visible mass of water droplets or ice crystals suspended near the Earth's surface. It reduces Visibility. Visibility refers to the distance one can see in the atmosphere, typically it measured horizontally. The reason why it's prediction is essential -

- Transportation safety - Fog prediction can reduce road accidents.
- Air travel - Prediction can manage Air travel much more effectively.
- Agriculture - Fog can affect the crop's health and management. Accurate forecast helps farmers plan their activities.

UNDERSTANDING DATA

Data was obtained from the Iowa State University. Iowa Environmental Mesonet website by choosing Kanpur as a selecting station and year range 2000 - 2023. The data is in Excel form and has a lot of parameters, but we selected some essential parameters so that qualitative input will be provided to our Models. Data is in a 30-minute interval. We decided to take Data from November till February for fog prediction; in this manner, the problem of Imbalance in data can be eradicated.

The fog prediction can be made in three ways-

- Classification - The aim is to find whether there is fog or no-fog.
- Multi-level Classification - Here, the output is Light fog, Heavy fog, or No fog.
- Regression - The outcome will be any number from 0-1 represents the fog index value, which shows the impact of fog.

I was instructed to work on the Regression problem. In this problem, there is further division based on the lag time-

- Short-term fog index prediction - This part aims to accurately predict the fog index of next 30 minute.
- Long-term fog index prediction - The aim is to predict the fog index of next 6 hours, one day, three days, and five days. So, this prediction model will give us an alert.

Specifically, I aim to improve the accuracy of this Long-term fog prediction. To do so, I did some preprocessing steps in which first, the data interval from 30 min is changed to the 6-hour duration interval, and then only winter months are selected for further process.

The parameters available in the 30-minute time duration dataset are as follows -

- Air Temperature
- Dew Point Temperature
- Relative Humidity in %
- Visibility in km
- Pressure

Missing values -

In the 30-minute interval dataset, some values of the columns, mainly belonging to the date interval 1 to 10 for almost all the months, are missing. So, due to the unavailability of data in this range, we need to drop the rows. In doing so, we get to know that the missing values are 40.15

METHODOLOGY

Problem statement - For long-term fog index prediction, I aim to increase the accuracy.

Evaluation metrics - I use the MSE (Mean Squared Error), and RMSE (Root Mean Squared Error) metrics to evaluate the results of the regression problem.

Preprocessing - We are here creating the model for the long term, so we need to resample the data. After resampling the obtained data contain these parameters -

- Average Air Temperature - This column is created by taking the average Air temperature value of 30-minute interval data.
- Average Dew Point Temperature - average dew point value of 30-minute interval data is used.
- Average Relative Humidity - calculated in the same way as the average dew point.
- Total Visibility in km - For this column, we added the visibility value for the 6-hour range in a 30-minute interval dataset.
- Average Pressure - Average is calculated in the same manner as the Air temperature.

Here, the target value is the fog index; the inclusion of this variable leads to the addition of some other derived parameters -

- Fog duration - For how long the fog occurs in a particular period.
- Energy loss -For this variable for visibility less than 2 km, the value is defined as

$$EnergyLoss = FogDuration(1 - e^{\frac{-0.053}{visibility}})$$

  and for visibility greater than 2 km, the energy loss is taken as zero.
- Fog index - Energy loss / total Fog duration of the 6 hour interval

One feature extraction column is Fog Month ,which has a value of 1 for December and January , and 0 for rest all. The whole 2000- 2023 dataset is split into train, validation and test dataset. The training dataset has data from year 2000 to 2015, the validation data timeline is 2016 to 2018, and the test dataset is purely unseen data by machine learning; its year value is from 2019 to 2023.

After pandas profiling, it was observed that the 30.9 % data of 6 hour interval is missing. To get rid of it, we drop the rows of missing data because most of them belongs to some specific duration (From 1 to 10 for almost all the month), for which the interpolation scheme will not work well.

Baseline persistence model

Simple a baseline implemented which is persistence model, where the assumption is that the current visibility condition will persist for a lead time of l. In other words, the model predicts Vl (and similarly, $Cl = Co$).

FOG MODELING

Persistence model - The persistence algorithm states that the value at the current timestep(t) is the same as the value at timestep $(t + 6h), (t + 24h), (t + 72h), (t + 120h)$.

Machine learning model - Different Machine learning models, Deep learning[3] (simple neural networks)[2],Time series models[4] like- Prophet, ARIMA and ARMA applies but only some of the Machine learning models gives better results . Those machine learning models are : Linear Regression[1], Random Forest Regression, Gradient Boosting, and AdaBoost; combining all the models and giving them equal weightage to find the output, the Voting Ensemble was introduced.

- **Linear Regression**

  Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables by fitting a linear equation to the observed data. The equation for simple linear regression is:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

  where:

  - Y is the dependent variable
  - X is the independent variable
  - β_0 is the y-intercept
  - β_1 is the slope (coefficient of the independent variable)

– ε is the error term

- Random Forest Regression -
  Random Forest Regression is an ensemble learning technique that can be used for regression tasks. It builds multiple decision trees during training and merges their predictions to obtain a more accurate and stable forecast.

- Gradient Boosting -
  Gradient Boosting is another ensemble learning technique used for regression tasks. It builds a series of weak learners (usually decision trees) sequentially, with each tree attempting to correct the errors of the combined ensemble so far.

- Adaboost -
  AdaBoost, short for Adaptive Boosting, is an ensemble learning technique . It builds a robust classifier by combining multiple weak classifiers sequentially.

- Voting Ensemble -
  Voting Ensemble for regression involves combining predictions from multiple regression models to create a more robust and accurate overall forecast.

So, by using the assumption that the prediction value may depend on its previous values, we have created three Models -

- **Base Model:-**
  Here, the selected machine learning models run for the dataset which obtained after the preprocessing step.

- **Model -1:-**
  In model 1, by considering the dependency of the output on its lag values, we add ten time lag value columns. Below example is provided which is used for 6-hour, one-day, three-day and five-day lead time prediction.
  Example: For lead k hour prediction, we add ten more columns as feature extraction, column-1 has a lag k hour target value, column -2 has a lag $2k$ hour target value and so on till column-10, which has a lag $10k$ hour target value.

- **Model -2:-**
  Model 2 is also based on the same approach that the dependency of the output is on its lag values; here, we add five more columns which contain the target value at the time in its previous days.
  Example: For the time To of Day Do, if I want to find the target value, then the five columns are added, and their values are like this: column -1 has a target value of time $To-24h$ (h stands for hours), column -2 contains the target value of time $To-48h$, continues till column -5 which have the target value of time $To-(24h*5)$.

EVALUATION CRITERIA

Mean Squared Error (MSE) -
MSE finds squared difference between actual and predicted value. Squaring prevents cancellation of negative terms. Below equation shows how to calculate MSE for n training examples wherein y_i is the actual value and $\hat{y}_i$ is the

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

RESULTS

Baseline persistence model result displayed below -

Time	6 Hour	1 Day	3 Day	5 Day
MSE	0.0209	0.0220	0.0272	0.0279

TABLE I. Baseline persistence model

Base model MSE (Mean Squared Error) results for different time lags.

Models	**Train**	**Validation**	**Test**
Linear Regression	0.0143	13.1074	13.1967
Random forest Regression	0.0018	0.0244	0.0205
Gradient Boost	0.0109	0.0243	0.0178
Adaboost	0.0005	0.0247	0.0192
Voting Ensemble	0.0045	0.8474	0.8631

TABLE II. Base model for 6 hour

Models	Train	Validation	Test
Linear Regression	0.0148	0.8308	0.8420
Random forest Regression	0.0021	0.0290	0.0224
Gradient Boost	0.0122	0.0325	0.0248
Adaboost	0.0007	0.0281	0.0203
Voting Ensemble	0.0051	0.0839	0.0820

TABLE III. Base model for 1 day

Models	Train	Validation	Test
Linear Regression	0.0166	0.0375	0.0263
Random forest Regression	0.0023	0.0432	0.0321
Gradient Boost	0.0141	0.0301	0.0218
Adaboost	0.0009	0.0362	0.0279
Voting Ensemble	0.0059	0.0316	0.0216

TABLE IV. Base model for 3 day

Models	Train	Validation	Test
Linear Regression	0.0174	0.0411	0.0353
Random forest Regression	0.0025	0.0388	0.0433
Gradient Boost	0.0148	0.0714	0.0823
Adaboost	0.0010	0.0377	0.0265
Voting Ensemble	0.0063	0.0376	0.0376

TABLE V. Base model for 5 day

Model-1 MSE (Mean Squared Error) result for different time lags.

Models	Train	Validation	Test
Linear Regression	0.0120	8.6358	8.6712
Random forest Regression	0.0014	0.0149	0.0125
Gradient Boost	0.0079	0.0133	0.0121
Adaboost	0.0003	0.0141	0.0114
Voting Ensemble	0.0019	0.0132	0.0112

TABLE VI. Model-1 for 6 hour

Models	Train	Validation	Test
Linear Regression	0.0139	1.4361	1.9522
Random forest Regression	0.0018	0.0265	0.0238
Gradient Boost	0.0104	0.0316	0.0254
Adaboost	0.0007	0.0254	0.0196
Voting Ensemble	0.0045	0.1276	0.1290

TABLE VII. Model-1 for 1 day

Models	Train	Validation	Test
Linear Regression	0.0163	0.0591	0.0562
Random forest Regression	0.0023	0.0408	0.0327
Gradient Boost	0.0131	0.0275	0.0218
Adaboost	0.0008	0.0325	0.0239
Voting Ensemble	0.0562	0.0310	0.0254

TABLE VIII. Model-1 for 3 day

Models	Train	Validation	Test
Linear Regression	0.0172	0.1074	0.1049
Random forest Regression	0.0024	0.0333	0.0410
Gradient Boost	0.0139	0.0587	0.0631
Adaboost	0.0010	0.0281	0.0276
Voting Ensemble	0.0059	0.0417	0.0434

TABLE IX. Model-1 for 5 day

Model-2 MSE (Mean Squared Error) results for different time lags.

Models	Train	Validation	Test
Linear Regression	0.0124	5.7204	5.7406
Random forest Regression	0.0015	0.0176	0.0155
Gradient Boost	0.0086	0.0155	0.0134
Adaboost	0.0004	0.0175	0.0137
Voting Ensemble	0.0037	0.3777	0.3839

TABLE X. Model-2 for 6 hour

Models	Train	Validation	Test
Linear Regression	0.0140	1.4260	1.4358
Random forest Regression	0.0019	0.0263	0.0236
Gradient Boost	0.0108	0.0239	0.0204
Adaboost	0.0007	0.0243	0.0201
Voting Ensemble	0.0046	0.1237	0.1248

TABLE XI. Model-2 for 1 day

Models	Train	Validation	Test
Linear Regression	0.0157	0.1414	0.1415
Random forest Regression	0.0022	0.0342	0.0279
Gradient Boost	0.0124	0.0286	0.0207
Adaboost	0.0008	0.0324	0.0223
Voting Ensemble	0.0053	0.0375	0.0319

TABLE XII. Model-2 for 3 day

Models	Train	Validation	Test
Linear Regression	0.0168	0.2045	0.2068
Random forest Regression	0.0023	0.0392	0.0375
Gradient Boost	0.0137	0.0875	0.0970
Adaboost	0.0010	0.0366	0.0291
Voting Ensemble	0.0058	0.0604	0.0610

TABLE XIII. Model-2 for 5 day

Lead Time	Base Model	Model-1	Model-2
6 Hour	0.0178	**0.0112**	0.0134
1 Day	0.0203	**0.0196**	0.0201
3 Day	0.0218	0.0218	**0.0207**
5 Day	0.0265	0.0276	0.0291

TABLE XIV. Summary comparison of Test MSE for Base model, Model-1, and Model-2 at different lead times

Graph between Actual output and Model-1 predicted output for 6 hour

Model-1 is trained on the train dataset with the help of Adaboost regression model for predicting the output value after 6 hour, the below graph shows that the how much close is the prediction from its actual value.
Blue color - Actual output
Red color - Predicted output

FIG. 1. Model-1

Graph between Actual output and Model-2 predicted output for 6 hour

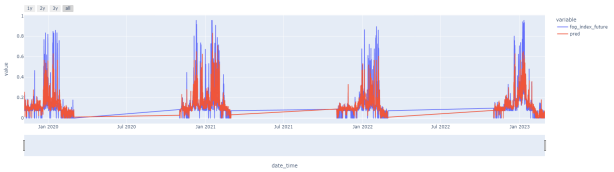

FIG. 2. Model-2

ANALYSIS AND DISCUSSION

The performance of the proposed fog prediction models is analyzed by comparing the Test Mean Squared Error (MSE) values obtained for different lead times, namely 6 hours, 1 day, 3 days, and 5 days. The comparison is carried out among the Base model, Model-1, and Model-2 to evaluate the impact of incorporating historical lag information into the prediction framework.

Short-term prediction (6-hour lead time)

For the 6-hour prediction horizon, a significant improvement is observed when lag-based models are employed. The Base model achieves a minimum Test MSE of 0.0178, whereas Model-1 reduces the error to 0.0112, corresponding to an improvement of approximately 37%. Model-2 also shows enhanced performance with a Test MSE of 0.0134, resulting in an improvement of nearly 25% compared to the Base model.

This substantial reduction in error indicates that short-term fog behavior strongly depends on its recent past values, and incorporating multiple lagged target variables enables the learning models to capture temporal persistence more effectively.

One-day prediction

For the one-day lead time, the improvement obtained by lag-based models is relatively moderate. The Base model yields a Test MSE of 0.0203, while Model-1 achieves a slightly lower error of 0.0196. Model-2 performs comparably with a Test MSE of 0.0201.

The reduced improvement suggests that as the prediction horizon increases, the influence of immediate past values weakens due to increasing atmospheric variability. Nevertheless, Model-1 still demonstrates marginal superiority over the Base model.

Medium-term prediction (3-day lead time)

In the case of 3-day prediction, Model-2 exhibits better performance compared to both the Base model and Model-1. While the Base model and Model-1 show similar Test MSE values around 0.0218, Model-2 reduces the error to 0.0207.

This behavior indicates that incorporating daily historical information, rather than closely spaced short-term lags, becomes more effective for medium-range forecasting. Hence, longer-term dependencies contribute more meaningfully at this prediction horizon.

Long-term prediction (5-day lead time)

For the 5-day lead time, all three modeling approaches demonstrate increased prediction error. The Base model attains a Test MSE of 0.0265, whereas Model-1 and Model-2 show slightly higher errors of 0.0276 and 0.0291, respectively.

The degradation in performance highlights the inherent difficulty of long-term fog prediction using limited surface meteorological parameters. As the lead time increases, uncertainty associated with atmospheric dynamics dominates, reducing the effectiveness of lag-based information.

CONCLUSION

From the comparative analysis, it is evident that incorporating historical fog index values significantly enhances prediction accuracy for short-term forecasting. Model-1, which utilizes multiple lagged values at fixed intervals, consistently performs best for short lead times, particularly for the 6-hour prediction.

Model-2, which incorporates daily lag dependencies, demonstrates improved performance for medium-term forecasting, especially for the 3-day lead time. However, for extended lead times such as 5 days, both lag-based approaches exhibit diminishing returns.

Overall, the analysis confirms that fog index prediction exhibits strong temporal dependence in the short and medium term, while long-term prediction remains challenging and may require additional atmospheric variables or advanced deep learning architectures for further improvement.

* manishac22@iitk.ac.in
† mkv@iitk.ac.in

[1] Andriy Burkov. *The hundred-page machine learning book*, volume 1. Andriy Burkov Quebec City, QC, Canada, 2019.

[2] Francois Chollet. Deep learning with python, vol. 1. *Greenwich, CT: Manning Publications CO*, 2017.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[4] Aurélien Géron Hands-On Machine Learning. with scikit-learn, keras, and tensorflow: Concepts, tools, and techniques to build intelligent systems, 2019.