Low Visibility Forecasting Using Numerical Weather Prediction Data

Topon Paul[1](✉), Vidhisha Reddy[2], Sai Prem Kumar Ayyagari[2], Ryusei Shingaki[1], Kaneharu Nishino[1], and Yoshiaki Shiga[1]

[1] Corporate Laboratory, Toshiba Corporation, Kanagawa 212-8582, Japan
{toponkumar.paul,ryusei1.shingaki}@toshiba.co.jp,
{kaneharu1.nishino,yoshiaki1.shiga}@toshiba.co.jp
[2] R&D Division, Toshiba Software (India) Pvt. Ltd., Bangalore 560034, KA, India,
{vidhisha.reddy,saipremkumar.ayyagari}@toshiba-tsip.com

Abstract. Low visibility is a critical factor affecting aviation and transportation safety, often leading to operational disruptions, delays, and potential hazards. Weather phenomena, such as fog, rain, and snow, significantly contribute to reducing visibility, making accurate prediction essential for mitigating risks. Conventional forecasting methods with time-series visibility and meteorological data often struggle with data imbalance and censored data issues, which impact forecasting accuracy, particularly in the low visibility range. In this paper, we propose a new approach by employing Censored Quantile Regression Neural Network and Light Gradient-Boosting Machine to forecast visibilities in the low and high visibility ranges and combining the forecast values by using a probabilistic classifier model built with Logistic Regression. We show the effectiveness of the proposed approach by performing experiments with two datasets of observed and forecast meteorological data from Japan and evaluating it in terms of forecasting errors and the accuracy of forecasting of low visibility. Experimental results suggest that our approach is well-suited to forecast low visibility with high accuracy up to 24 hours ahead.

Keywords: Visibility forecasting · Censored data · Imbalanced data · Meteorological data.

1 Introduction

Visibility, a fundamental meteorological parameter, refers to the distance at which an object or light can be clearly perceived by the human eye in the atmosphere. Low visibility is a critical factor affecting aviation and transportation safety, often leading to operational disruptions, delays, and potential hazards. Weather phenomena, such as fog, heavy precipitation, snow, dust, and smoke, significantly contribute to reducing visibility, making accurate prediction of low visibility essential for mitigating risks. The definition of low visibility depends on the application; for example, in aviation, it is in kilometer range while in road transportation, it is in meter range.

There have been proposed a number of methods for visibility forecasting using time-series visibility datasets with weather data in literature. These methods include regression-based methods, deep learning-based methods, and physical models, which combine pre-processing of visibility data, addition of features required for better prediction, and selection of important features. For example, in [11], XGBoost and Light Gradient-Boosting Machine (LightGBM) are used to train a multimodal fusion model for visibility prediction. In [12], six machine learning methods: linear discriminant analysis, decision tree, Naïve Bayes, linear SVM, kNN, and neural network are used for visibility prediction. In [4], the authors have used two deep learning models: a Multilayer Perceptron (MLP) and a Convolutional Neural Network (CNN) to forecast one-step ahead visibility; they have concluded that the forecasting errors are affected by the lagged values used in building the forecasting models using the methods. In [13], the authors have used MLP to create 28 different prediction models of dominant visibility; these models are designed to forecast dominant visibility using different combinations of historical visibility data over various time-periods. They have concluded that inclusion of other meteorological data can stabilize forecasting accuracy. The authors in [9] have used a forward feature selection algorithm based on evolutionary computation to determine the optimal meteorological variables and deep learning as well as conventional machine learning methods to build forecasting models. In [5], the authors have utilized deep learning and conventional regression-based methods to forecast peak values accurately by utilizing the learning capabilities of these methods on time-series data. However, these traditional prediction models often struggle with censored (clipped) data, imbalanced data, and peak-shift issues, which impact forecasting accuracy, particularly in the low visibility range.

To handle censored data effectively, several state-of-the-art methodologies, such as Tobit exponential smoothing, quantile regression neural networks, and censored regression models, have been proposed to enhance prediction accuracy of a regression model. In [8], the authors have proposed Tobit Exponential Smoothing (Tobit ETS), an enhancement of traditional time-series forecasting for censored observations; this method incorporates Kalman filtering to dynamically update state vectors based on observed values and imposes censoring thresholds. In [7], the authors have proposed Censored Quantile Regression Neural Networks (CQRNN), a neural network-based learning framework designed to estimate conditional quantiles in censored datasets. Unlike traditional regression methods, CQRNN provides a fuller representation of the distribution of the target variable. In [2], the authors have extended censored regression methodologies by introducing a penalized Tobit likelihood approach to handle high-dimensional censored data. It employs Generalized Coordinate Descent (GCD) to iteratively optimize the Tobit likelihood function, and a soft-thresholding rule based on quadratic majorization to minimize penalized loss functions. In [1], the authors have applied Multi-Output Censored Quantile Regression Neural Networks (Multi-CQNN) to mobility demand forecasting; they have integrated Bayesian modeling for improved interpretability. In it, a multi-output CQNN is trained with an asymmetric Laplace likelihood function to model censored data,

and the network applies a tilted loss function to refine predictions under censoring constraints. The major problem of these approaches is that they can handle censored data properly but in the case of visibility forecasting, their forecast values are biased toward high visibility ranges due to imbalanced data.

To address these issues, one attempt is made in [6] by employing multiple forecasting models; however, though the method obtains better forecasting accuracy in the low visibility range, the forecasting accuracy in the high visibility range is low, making many false alarms. Moreover, in it, the weights of low and high visibility ranges are set manually; determination of appropriate weights at various observation stations might be difficult.

1.1 Main Contributions

To overcome the limitations of existing technologies to handle imbalanced and censored data properly and forecast low visibility with high accuracy, we propose a new approach by combining forecasting methods CQRNN and LightGBM with a probabilistic model employing Logistic Regression after selecting and transforming relevant meteorological variables. This new approach aims to enhance forecasting accuracy by leveraging advanced machine learning techniques and optimizing feature selection and provides a more comprehensive assessment of extreme low visibility events. We show the effectiveness of the proposed approach by performing experiments with two datasets of observed weather data and Numerical Weather Prediction (NWP) data, sourced from Japan Meteorological Agency (JMA).

2 Methods

In our proposed approach, to deal with the censored data and imbalanced data, we learn two forecasting models by employing CQRNN, and LightGBM with Kernel Density Estimation (KDE) weights. The issue of censored data is dealt with CQRNN, and the issue of imbalanced data is dealt with LightGBM with KDE weights. These models are tuned in such a way that CQRNN focuses in the low visibility range while LightGBM with KDE weights focuses in both visibility ranges. The forecast values by these two models are ensembled through Logistic Regression classifier, which gives the probabilities of visibility in the low and high ranges given the observed and forecast weather data. The outline of the proposed approach is shown in Fig. 1.

2.1 Selection of Meteorological Variables

Selection of the most relevant meteorological variables is very important because the irrelevant variables act as noises to input data and reduce the forecasting accuracy.

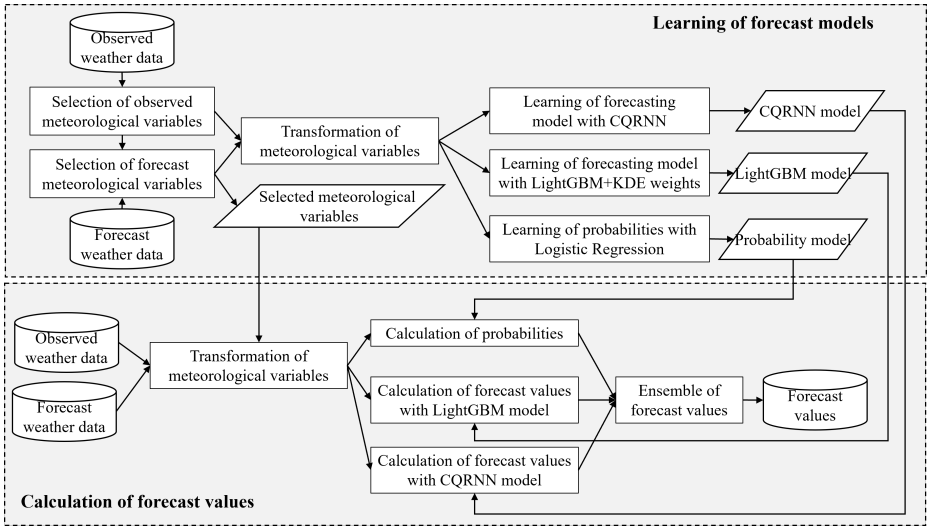


Fig. 1: Outline of proposed approach

Selection of Observed Meteorological Variables To select the observed meteorological variables, we use the correlation coefficient between a meteorological variable and the target variable (visibility). Meteorological variables having higher correlation coefficients are selected. Then, correlation coefficients among the meteorological variables are calculated and only one from each group of highly correlated variables is selected. For example, sunshine hours and solar radiation are highly correlated; in this case, only sunshine hours is selected and solar radiation is discarded.

Selection of Forecast Meteorological Variables Instead of using correlation between the target variable (visibility) and a meteorological variable, we use the quality of the forecast data as a selection criterion. If the quality of the forecast data is bad, the forecasting accuracy will fall. We use the correlation between the observed and forecast data of a meteorological variable as the selection criterion. The higher the correlation coefficient, the higher the quality of the meteorological variable. Meteorological variables having higher correlation coefficients are selected.

2.2 Transformation of Meteorological Variables

The transformation of meteorological variables is applied to improve the quality of the data and make it more suitable for predictive modeling. By normalizing and stabilizing the variance of the variables, we aim to enhance the performance of our models and obtain more accurate predictions.

To transform Relative Humidity (RH) and Precipitation (PR), a square root transformation is applied. This transformation is intended to normalize the data

and reduce skewness; it also makes the relative humidity linear to visibility. The transformed relative humidity (RH_{tr}) and precipitation (PR_{tr}) are as follows:

$$RH_{tr} = \sqrt{100 - RH}; \tag{1}$$
$$PR_{tr} = \sqrt{100 - PR}. \tag{2}$$

Pressure variables are transformed by calculating the difference between Surface Pressure (SP, station pressure)/Vapor Pressure (VP) and Sea Level Pressure (SLP). The Differential Surface Pressure (DSP), and the Differential Vapor Pressure (DVP) are calculated as follows:

$$DSP = SP - SLP; \tag{3}$$
$$DVP = VP - SLP. \tag{4}$$

Transformation of Visibility Variable Since the visibility values are censored, and low visibility values are rare events, logit transformation of the target variable might be helpful to build a stable model and to handle rare events. The steps to transform the target variable (visibility, y) using logit transformation are as follows.

i) Find y_{min} and y_{max}: determine the minimum (y_{min}) and maximum (y_{max}) values of the target variable.
ii) Adjust y_{min} and y_{max}: adjust the minimum and maximum values slightly to avoid boundary issues:

$$y_{min} = y_{min} - 0.0001; \tag{5}$$
$$y_{max} = y_{max} + 0.0001. \tag{6}$$

iii) Scale the target variable: scale the target variable y using the formula:

$$y_s = (y - y_{min})/(y_{max} - y_{min}). \tag{7}$$

iv) Perform logit transformation: apply the logit transformation to the scaled variable:

$$logit(y_s) = \log\left(\frac{y_s}{1 - y_s}\right) \tag{8}$$

During calculation of forecast values, a forecast value (y_p) is transformed back to the original scale using the following steps.

i) Get the adjusted y_{min} and y_{max} values, calculated during transformation of the target variable using logit transformation.
ii) Apply the reverse logit transformation to the forecast value:

$$y_l = \frac{\exp(y_p)}{1 + \exp(y_p)}. \tag{9}$$

iii) Convert the reverse logit value back to the original scale:

$$y_f = y_{min} + y_l * (y_{max} - y_{min}). \tag{10}$$

2.3 Learning of Models

Here, we describe the learning of various forecasting models as well as the probabilistic model. The input and output data to these models are shown in Fig. 2. The input data consists of lagged values of the observed meteorological variables and the values of forecast meteorological variables up to the forecast horizon. The output data are the lead values of the target variable up to the forecast horizon. The pseudocode of learning of forecast models is given in Algorithm 1.

Algorithm 1: Pseudocode of learning of forecast models

Data: Observed weather data: $\boldsymbol{X}_o(t)$, observed data of target variable: $\boldsymbol{y}(t)$, forecast weather data: $\boldsymbol{X}_f = [\boldsymbol{X}_f(t+1), \ldots, \boldsymbol{X}_f(t+\Delta)]$

Result: $Model_{\text{CQRNN}}$, $Model_{\text{LightGBM}}$, $Model_{\text{prob}}$, NP

// l: lag size, Δ: forecast horizon, lv: threshold of LV

Transform meteorological variables of $\boldsymbol{X}_o(t)$ and $\boldsymbol{X}_f$;

Normalize $\boldsymbol{X}_o(t)$, $\boldsymbol{X}_f$, $\boldsymbol{y}(t)$ and get normalization parameters (NP) ;

Create lag values of $\boldsymbol{X}_o(t)$: $\boldsymbol{X}_{ol} = [\boldsymbol{X}_o(t-l), \ldots, \boldsymbol{X}_o(t-1), \boldsymbol{X}_o(t)]$;

Create lag values of $\boldsymbol{y}(t)$: $\boldsymbol{Y}_l = [\boldsymbol{y}(t-l), \ldots, \boldsymbol{y}(t-1), \boldsymbol{y}(t)]$;

Set $Model_{\text{CQRNN}} = \{\}, Model_{\text{LightGBM}} = \{\}, Model_{\text{prob}} = \{\}$;

for $s \in \{1, 2, \ldots, \Delta\}$ **do**

  Create lead values of $\boldsymbol{y}(t)$: $\boldsymbol{y}_s = \boldsymbol{y}(t+s)$;

  Extract forecast weather data: $\boldsymbol{X}_{fs} = \{\boldsymbol{X}_f(t+1), \ldots, \boldsymbol{X}_f(t+s)\}$;

  Create model input data by merging: $\boldsymbol{X}_s = [\boldsymbol{X}_{ol}, \boldsymbol{Y}_l, \boldsymbol{X}_{fs}]$;

  Calculate KDE weights $\boldsymbol{w}_s$ of $\boldsymbol{X}_s$ using $\boldsymbol{y}_s$;

  Learn models: $mc = CQRNN(\boldsymbol{X}_s, \boldsymbol{y}_s)$,
  $ml = LightGBM(\boldsymbol{X}_s, \boldsymbol{y}_s, \boldsymbol{w}_s)$,
  $pm = LogisticRegression(\boldsymbol{X}_s, \boldsymbol{y}_s > lv)$;

  Append models to $Model_{\text{CQRNN}}$, $Model_{\text{LightGBM}}$, $Model_{\text{prob}}$;

end

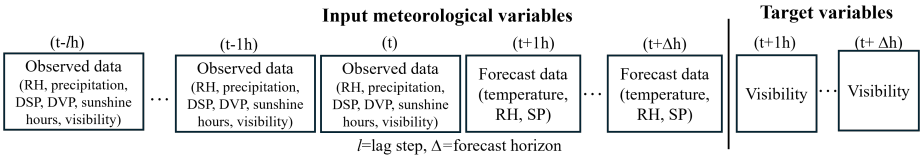


Fig. 2: Illustration of input and output data of various models

Learning of Forecasting Model with CQRNN Among the regression models proposed to deal with censored data, we select CQRNN because it is a non-parametric regression model and can capture non-linear relationships among the

variables. The CQRNN model estimates the conditional quantiles of the target variable given the input variables. Unlike Ordinary Least Squares (OLS) regression, which focuses only on estimating the mean of the target variable, quantile regression provides insights into the entire distribution. By predicting different quantiles, CQRNN enables a more comprehensive understanding of uncertainty and variability in the target variable. Fig. 3 illustrates the forecast values of visibility at various quantiles. In CQRNN, the model is initialized with multiple quantiles to estimate, a neural network is trained using a loss function that accommodates both observed and censored data, and censored observations are re-weighted iteratively based on predicted quantiles to refine estimates.

Instead of relying on predefined weights as in [6], the introduction of CQRNN provides a data-driven approach to estimate different quantile predictions. Now, by leveraging the 0.1-quantile predictions from the CQRNN model, the approach ensures that the model captures lower-bound estimates (low visibility) effectively. This modification enhances the model's ability to account for uncertainty, leading to more reliable and flexible predictions.

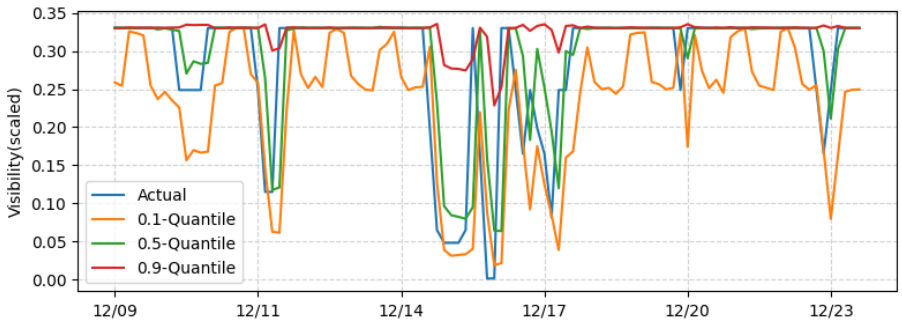


Fig. 3: Illustration of forecasting by CQRNN at various quantiles

Learning of Forecasting Model with LightGBM+KDE Weights The problem at hand revolves around the challenge posed by imbalanced data on the performance of machine learning algorithms. Imbalanced data occurs when certain classes or categories within the dataset are under-represented, often leading to sub-optimal model performance, especially in cases where the focus is on rare occurrences within the data. To deal with this issue, we build another forecast model with LightGBM [3] with KDE weights[10] assigned to the input data.

LightGBM is a regression method based on decision trees and XGBoost. It utilizes various characteristics, such as sparse optimization, parallel training, multiple loss functions, regularization, bagging, and early stopping of XGBoost. It has faster training speed and better accuracy than other gradient boosting frameworks.

The steps to calculate KDE-based weights to the input data are as follows.

i) Determining density function of the target variable: To address the issue of imbalanced data, the first step involves calculating a measure of rarity for the target variable. This is achieved by determining the density function ($p(y)$) of the target variable. The density of the data points is computed using the KDE method:

$$p(y) = \frac{1}{Nh} \sum_{i-1}^{N} K\left(\frac{y - y_i}{h}\right) \tag{11}$$

where N is the number of visibility values, h is the bandwidth of KDE, and K is the kernel function (Gaussian distribution).

ii) Normalizing density points: The density points obtained in the previous step are then normalized using the min-max normalization technique:

$$p\prime(y) = \frac{p(y) - \min(p(Y))}{\max(p(Y)) - \min(p(Y))} \tag{12}$$

where Y is the set of N visibility values in the dataset.

iii) Assigning sample weights: The final step involves assigning sample weights ($weight(y)$) based on the normalized density points as follows.

$$weight(y) = \max(1 - \alpha p\prime(y), \epsilon) \tag{13}$$

where $\alpha(0 \leq \alpha \leq 1.0)$ is a user-defined parameter that controls the influence of the density points on the sample weights; a lower value of α assigns greater importance to rare occurrences, while a higher value prioritizes common occurrences. $\epsilon(0 \leq \epsilon \leq 1.0)$ is a user-defined constant value; it serves as a threshold to ensure that the adjusted sample weight does not drop below a certain minimum value, thereby preventing excessively low weights. In this paper, we set α and ϵ to 0.7, and 0.4, respectively, based on some preliminary experiments, which ensures that the model learned by LightGBM with KDE weights focuses in both regions of visibility.

Learning of Probabilities with Logistic Regression The probabilities of low and high visibility are calculated by using Logistic Regression (Logit Regression) classifier. This model is chosen for its effectiveness in binary classification tasks, where the goal is to categorize input data into one of two groups: low or high visibility range. Here, group 0 represents instances where visibility is less than or equal to lv (threshold of low visibility), while group 1 represents instances where visibility exceeds lv. The low visibility threshold is specific to a dataset; in this paper, we set lv to 1 km which is the threshold of visibility distance for fog. The classifier assigns probabilities to each group based on the input data, indicating the likelihood of belonging to each group. These probabilities are then utilized in calculation of forecasting values. By considering the probabilities assigned to each group, the model can make more informed forecasting, considering the uncertainty inherent in the classification task.

2.4 Calculation of Forecast Values

The pseudocode of calculation of forecast values is given in Algorithm 2. Given the observed and forecast meteorological variables, the variables are transformed and then forecast values are calculated with the CQRNN model and LightGBM model, and probabilities of low and high visibility is calculated with the learned Logistic Regression model.

Algorithm 2: Pseudocode of calculation of forecast values

Data: $Model_{\text{CQRNN}}$, $Model_{\text{LightGBM}}$, $Model_{\text{prob}}$, NP,
observed weather data: $X_{ol} = [X_o(t-l), \ldots, X_o(t-1), X_o(t)]$,
observed data of target variable: $Y_l = [y(t-l), \ldots, y(t-1), y(t)]$,
forecast weather data: $X_f = [X_f(t+1), \ldots, X_f(t+\Delta)]$
// l: `lag size`, Δ: `forecast horizon`
Result: Forecast values $\hat{\boldsymbol{y}}_f$

Transform meteorological variables of X_{ol} and X_f;
Normalize X_{ol}, X_f, Y_l using normalization parameters (NP);
Set $\hat{\boldsymbol{y}}_f = \{\}$;
for $s \in \{1, 2, \ldots, \Delta\}$ **do**
  Extract forecast weather data $X_{fs} = [X_f(t+1), \ldots, X_f(t+s)]$;
  Create model input data by merging: $X_s = [X_{ol}, Y_l, X_{fs}]$;
  Calculate forecast values: $\hat{y}_{CQRNN} = Model_{\text{CQRNN},s}(X_s)$,
  $\hat{y}_{LightGBM} = Model_{\text{LightGBM},s}(X_s)$;
  Caculate probabilities: $p_l, p_h = Model_{\text{prob},s}(X_s)$;
  Calculate $\hat{y}_s = \hat{y}_{CQRNN} \times p_l + \hat{y}_{LightGBM} \times p_h$;
  Transform $\hat{y}_s$ to original scale by using NP and append it to $\hat{\boldsymbol{y}}_f$;
end

Then, the final forecast value ($\hat{y}$) is calculated though the ensemble of the two forecast values as follows:

$$\hat{y} = \hat{y}_{CQRNN} \times p_l + \hat{y}_{LightGBM} \times p_h \tag{14}$$

where $\hat{y}_{CQRNN}$ and $\hat{y}_{LightGBM}$ are the forecast values calculated by employing the CQRNN model and LightGBM model, respectively, p_l and $p_h (= 1 - p_l)$ are the probabilities of the low and high visibility, respectively. The rationale behind assigning the probability of low visibility to the forecast value by the CQRNN model and that of high visibility to the forecast value by the LightGBM model is as follows. Since the CQRNN model with a 0.1 quantile is specifically designed to predict the lower quantile of the distribution, it focuses on estimating the lowest range of the target variable. That means CQRNN model makes a prediction for low visibility; that is why the probability of low visibility is assigned to the forecast value by the CQRNN model. On the other hand, when the probability of high visibility is higher, we want to a make a prediction by emphasizing more on the forecast values by the LightGBM model because it is tuned to make better prediction in the high visibility range than CQRNN; that is why the probability of high visibility is assigned to the forecast values by the LightGBM model.

2.5 Evaluation Metrics

To evaluate the forecasting methods, we use forecasting errors: RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) in the low visibility range and the accuracy of classification (F1 score, AUC) of visibilities in the low and high ranges. The lower the RMSE and MAE values, the better the performance of the forecasting model while the higher the F1 score and AUC, the better the performance of the forecasting model.

3 Experiments and Results

3.1 Datasets

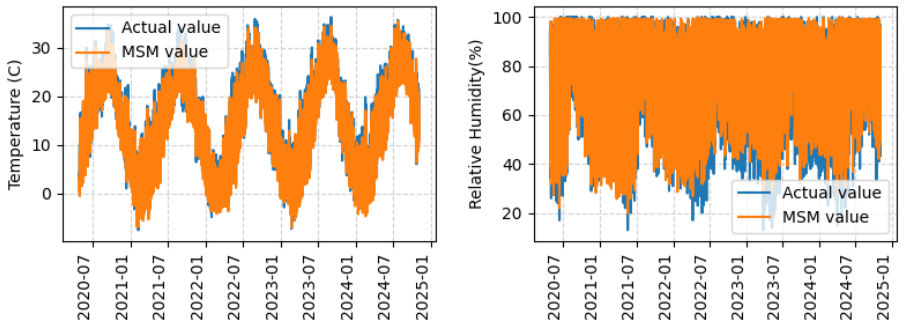
To show the effectiveness of the proposed approach, we perform experiments with two datasets with observed and forecast weather data provided by JMA at Sendai and Akita weather stations in Miyagi and Akita Prefectures in Japan. We selected these two weather stations because they have diverse climatic conditions due to their geographical locations. The observed weather data is publicly available for download at the JMA website: `https://www.data.jma.go.jp/risk/obsdl/index.php`. The data is recorded at hourly interval. The observed meteorological variables are visibility, relative humidity, precipitation, temperature, dew-point temperature, vapor pressure, sea level pressure, surface pressure (station pressure), wind speed, snow cover, solar radiation, sunshine hours, and weather condition (sunny, cloudy, rainy, etc.).

As forecast weather data, we use Meso-Scale Model (MSM) forecast data of numerical weather prediction provided by JMA. The horizontal resolution of the MSM is 5km, and it provides 39-hour ahead forecast at 00, 03, 06, 09, 12, 15, 18, 21 UTC. The data is recorded at 3-hour interval with hourly interval in the forecast horizon. The forecast meteorological variables are pressure reduced to mean sea level, surface pressure, U and V components of wind, temperature, relative humidity, low cloud cover, high cloud cover, total cloud cover, accumulated precipitation, downward shortwave radiation flux. Temperature is converted from Kelvin scale to Celsius scale. We collect the data of both observed and forecast meteorological variables during the period 2020-2024. We use the data of 2020-2022 as the training data and 2023-2024 as the evaluation data.

3.2 Selected Meteorological Variables

Observed Meteorological Variables From the observed weather data, we select precipitation, differential surface pressure, differential vapor pressure, sunshine hours and relative humidity having higher correlation with visibility.

Forecast Meteorological Variables Using correlation coefficients (CC) of the actual and forecast meteorological variables, temperature, relative humidity, and surface pressure (see Fig. 4) are selected from the MSM forecast data as their CC are (0.96, 0.66, 0.95) and (0.97, 0.62, 0.80) respectively in the Sendai and Akita datasets, with all CC being greater than 0.50.



(a) Temperature (b) Relative humidity

Fig. 4: Selected two meteorological variables from MSM forecast data

3.3 Experimental Setup

We perform experiments using various forecasting methods for forecast horizon of 1, 3, and 24 hours. As a baseline method, we use LightGBM. For LightGBM, we use the defaults settings of the parameters of the python package. For CQRNN, the settings of the parameters are as follows: quantiles = [0.1, 0.2, 0.5, 0.7, 0.9, 1.0]; number of epochs = 50; batch size = 128; number of hidden units=100; and activation=relu. For CQRNN, we use 0.1-quantile forecast values. For Sendai dataset, we use the lag step of 6 for all observed variables; for Akita, it is set to 1. Logistic Regression is learned with balanced class weight.

3.4 Experimental Results

Comparison between Baseline Method and Proposed Approach First, we perform experiments with the lag values of the selected observed meteorological variables after transforming the values and applying the baseline method and the proposed approach. The experimental results are shown in Table 1. From the evaluation scores, we find that the proposed approach is consistently better than the baseline method in all evaluation metrics. In terms of accuracy of forecasting the values in the low and high visibility ranges, the proposed approach is significantly better.

Effects of Using Forecast Data It is expected that by using forecast meteorological variables, the performance of the forecasting methods might improve by mitigating the problem of peak shift. However, since the quality of the forecast data might affect the performance of forecasting of visibility, here, we use the lead values of the observed meteorological variables as the forecast data. We perform experiments with the lag values of the selected observed meteorological variables merged with the forecast data created in this way. Here, we do not

Table 1: Comparative results of baseline method and proposed approach

Dataset	Forecast step	Baseline Method				Proposed Approach			
		RMSE	MAE	F1 score	AUC	RMSE	MAE	F1 score	AUC
Sendai	1	5.18	3.59	0.20	0.56	**4.42**	**2.36**	**0.51**	**0.69**
	3	7.22	6.22	0.10	0.53	**5.71**	**3.99**	**0.32**	**0.63**
	24	14.29	13.94	0.01	0.50	**13.10**	**12.26**	**0.05**	**0.52**
Akita	1	9.76	8.26	0.13	0.53	**5.89**	**3.32**	**0.37**	**0.75**
	3	11.64	10.39	0.04	0.51	**7.09**	**4.65**	**0.28**	**0.66**
	24	14.82	14.35	0.00	0.50	**9.87**	**8.20**	**0.15**	**0.56**

Table 2: Comparative results of the proposed approach under various conditions of input data to forecast models

Dataset	F. Step	Lag data only				Lag and forecast data				Target transformation			
		RMSE	MAE	F1 score	AUC	RMSE	MAE	F1 score	AUC	RMSE	MAE	F1 score	AUC
Sendai	1	4.42	2.36	0.51	0.69	**3.17**	**1.84**	**0.68**	**0.77**	4.56	2.25	0.53	0.72
	3	5.71	3.99	0.32	0.63	**3.72**	2.68	**0.37**	**0.66**	4.46	**2.63**	0.33	0.63
	24	13.10	12.26	0.05	0.52	**5.45**	**4.53**	**0.22**	**0.57**	7.84	6.12	0.15	0.55
Akita	1	5.89	3.32	0.37	0.75	**1.42**	0.85	**0.49**	0.81	1.52	**0.71**	0.48	**0.86**
	3	7.09	4.65	0.28	0.66	**1.91**	1.22	**0.43**	0.75	2.24	**1.21**	0.42	**0.79**
	24	9.87	8.20	0.15	0.56	**3.67**	**2.67**	0.29	0.62	5.65	3.47	**0.34**	**0.66**

use the transformed values of the target variable. The experimental results are shown in Table 2. From the evaluation scores, we find that by adding forecast data, the performance of the proposed approach is significantly improved.

Effects of Transformation of Target Variable Next, we investigate whether transformation of the target variable improves the performance of forecasting approach or not. In this case, we perform experiments with transformation of the target variable. The experimental results are shown in Table 2. From the experimental results, we find that though transformation of the target variable produces better AUC on the Akita dataset, its performance in terms of other evaluation metrics is worse. Moreover, it produces worse scores on the Sendai dataset. Therefore, it is better not to transform the target variable.

Effects of Using MSM Forecast Data In the previous subsection, we have observed that by adding the lead values of the observed meteorological variables as the forecast data improves forecasting performance significantly; however, in the real world, we must use real forecast data. In this context, we perform experiments with MSM forecast data instead of the lead values. The experimental results are shown in Table 3. From the experimental results, we find that using MSM forecast data instead of the lead values results in slightly lower evaluation

Table 3: Effects of using MSM forecast data

Dataset	Forecast step	Lag+lead forecast data				Lag+MSM forecast data			
		RMSE	MAE	F1Score	AUC	RMSE	MAE	F1Score	AUC
Sendai	1	**3.17**	**1.84**	**0.68**	**0.77**	4.64	2.58	0.57	0.72
	3	**3.72**	**2.68**	**0.37**	**0.66**	5.94	4.18	0.32	0.64
	24	**5.45**	**4.53**	**0.22**	**0.57**	10.83	9.78	0.05	0.52
Akita	1	**1.42**	**0.85**	**0.49**	**0.81**	5.48	3.16	0.39	0.74
	3	**1.91**	**1.22**	**0.43**	**0.75**	6.32	4.02	0.32	0.69
	24	**3.67**	**2.67**	**0.29**	**0.62**	8.79	7.45	0.13	0.56

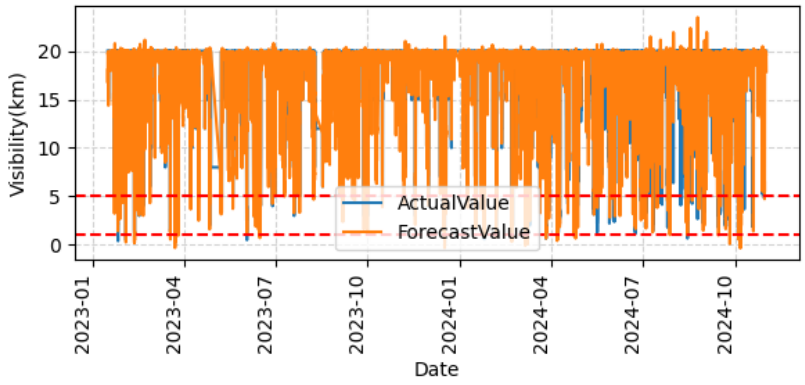
scores, which is expected because the quality of MSM forecast data is inferior to lead values of the variables. The forecast values using observed and forecast data of meteorological variables are shown in Fig. 5. From the experimental results, we find that for short-term forecasting, such as rolling 3-hour forecast, the proposed approach produces forecast values with very high accuracy. As the forecasting horizon becomes longer, the forecasting accuracy falls, and the forecast values shift away from the actual values.

Forecasting of Visibility Under Various Weather Conditions Next, we investigate the performance of the proposed approach under various weather conditions: fog, snow, and rain at Sendai weather station. Here, we perform experiments with the proposed approach by using lag values of the selected observed meteorological variables and MSM forecast data. We focus on those days (11 days) when visibility drops below 1 km. The weather conditions on three of those days are shown in Table 4. The forecast values on those selected days are shown in Figs. 6, 7, and 8. In the figures, 3-hour rolling forecast means the forecasting is scheduled at 00:00, 03:00, 06:00, 09:00, 12:00, 15:00, 18:00, 21:00 and at each scheduling time, values are forecast for 1-3 hours ahead. Similarly, in the case of 24-hour rolling forecast, forecasting is made at 00:00 every day.

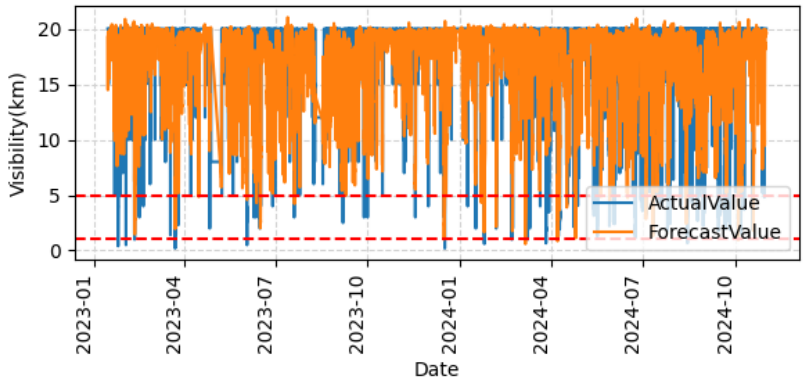
Form the experimental results of 3-hour rolling forecast, we find that the proposed method can produce forecast values with very high accuracy. In the case of 24-hour rolling forecast, as the forecasting horizon increases, the forecasting accuracy falls; however, using these forecast values it is possible to issue warning by the transportation and aviation authorities that in the following days visibility will drop below 5km, which might be helpful for the flight operators to reschedule their flights.

Table 4: Weather condition during poor visibility

Timestamp	Visibility(km)	Weather	Timestamp	Visibility(km)	Weather
2023/3/23 3:00	0.2	Fog	2023/6/2 18:00	0.5	Rain
2023/3/23 6:00	0.3	Fog	2024/2/21 12:00	1.0	Snow



(a) 3-hour rolling forecast



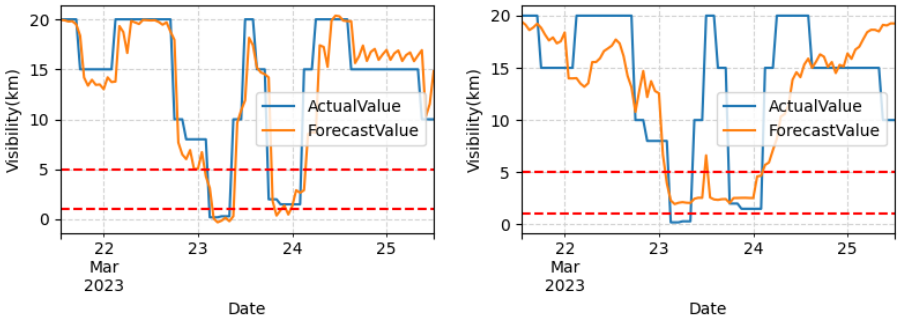
(b) 24-hour rolling forecast

Fig. 5: Forecasting using observed and forecast data of meteorological variables

Execution Time The training time of a 3-hour ahead forecast model is around 144 seconds, and the time required to calculate forecast values is 159 milliseconds on a computer with Intel® Core™ i7-1165G7@2.80GHz processor and 16GB of RAM running on Windows 11 Pro operating system.

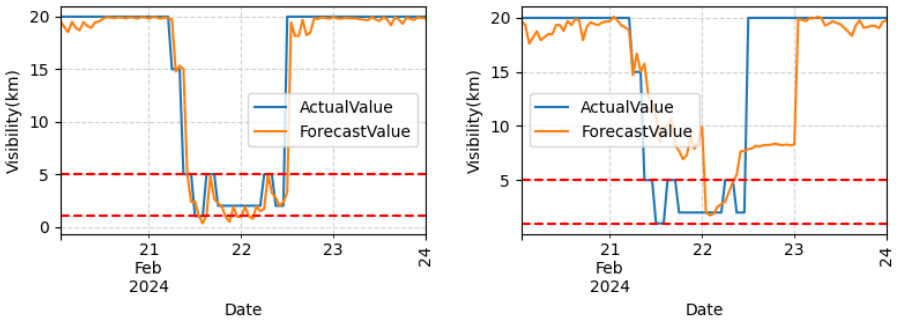
4 Conclusion

In this paper, we have proposed a new approach to forecast low visibility with high accuracy and shown the effectiveness of the proposed approach by performing experiments with two datasets from Japan. From the experimental results, we have found that our proposed approach is better than the baseline method and can produce forecast values with very high accuracy for short-term forecasting, and for mid-term forecasting, it is possible to tell the trend of low visibility



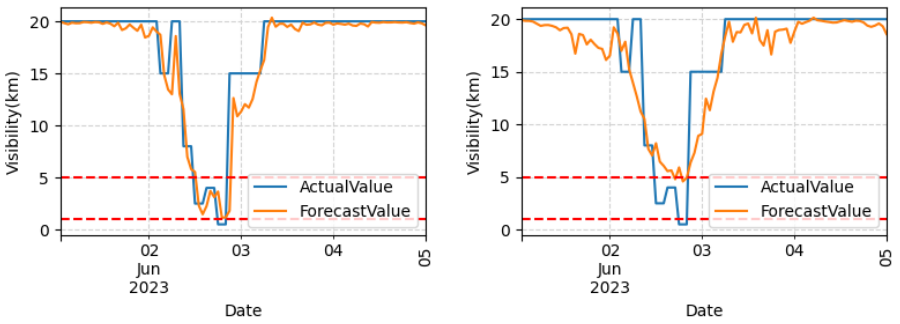
(a) 3-hour rolling forecast (b) 24-hour rolling forecast

Fig. 6: Forecasting of low visibility caused by fog



(a) 3-hour rolling forecast (b) 24-hour rolling forecast

Fig. 7: Forecasting of low visibility caused by snow



(a) 3-hour rolling forecast (b) 24-hour rolling forecast

Fig. 8: Forecasting of low visibility caused by rain

in the next day, which might be helpful for the transportation and aviation authorities to adjust their operation schedules.

Since we have confirmed the performance of the proposed approach using real-world datasets, we want to incorporate it into a traffic control system and perform field test in coming days. Moreover, we want to investigate the acceptable level of accuracy for forecasting of low visibility in various real-world applications in our future work.

References

1. Huttel, F., Peled, I., Rodrigues, F., Pereira, F.: Modeling censored mobility demand through censored quantile regression neural networks. IEEE Transactions on Intelligent Transportation Systems **23**(11), 21753–21765 (2022)
2. Jacobson, T., Zou, H.: High-dimensional censored regression via the penalized tobit likelihood. Journal of Business & Economic Statistics **42**(1), 286–297 (2024)
3. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, P., Ma, P., Ye, Q., Liu, T.Y.: LightGBM: a highly efficient gradient boosting decision tree. In: NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 3149–3157 (2017)
4. Ortega, L.C., Otero, L.D., Solomon, M., Otero, C.E., Fabregas, A.: Deep learning models for visibility forecasting using climatological data. International Journal of Forecasting **39**(2), 992–1004 (2023)
5. Paul, T., Raghavendra, S., Ueno, K., Ni, F., Shin, H., Nishino, K., Shingaki, R.: Forecasting of reservoir inflow by the combination of deep learning and conventional machine learning. In: 2021 International Conference on Data Mining Workshops (ICDMW). pp. 558–565 (2021)
6. Paul, T., Reddy, V., Ayyagari, S.P.K., Shingaki, R., Nishino, K.: Forecasting of low visibility using weather and air quality data for safe and smooth transportation operation. In: Proceedings of ICMLA 2024 : 23rd International Conference on Machine Learning and Applications (2024)
7. Pearce, T., Jeong, J.H., Jia, Y., Zhu, J.: Censored quantile regression neural networks for distribution-free survival analysis. In: Advances in Neural Information Processing Systems, NeurIPS (2022)
8. Pedregal, D.J., Trapero, J.R., Holgado, E.: Tobit exponential smoothing, towards an enhanced demand planning in the presence of censored data (2024), `https://arxiv.org/abs/2407.17920`
9. Peláez-Rodríguez, C., Pérez-Aracil, J., Casanova-Mateo, C., Salcedo-Sanz, S.: Efficient prediction of fog-related low-visibility events with machine learning and evolutionary algorithms. Atmospheric Research **295**, 106991 (2023)
10. Steininger, M., Kobs, K., Davidson, P., Krause, A., Hotho, A.: Density-based weighting for imbalanced regression. Machine Learning **110**, 2187–2211 (2021)
11. Zhang, C., Wu, M., Chen, J., Chen, K., Zhang, C., Xie, C., Huang, B., He, Z.: Weather visibility prediction based on multimodal fusion. IEEE Access **7**, 74776–74786 (2019)
12. Zhang, Y., Wang, Y., Zhu, Y., Yang, L., Ge, L., Luo, C.: Visibility prediction based on machine learning algorithms. Atmosphere **13**(7) (2022)
13. Zhu, L., Zhu, G., Han, L., Wang, N.: The application of deep learning in airport visibility forecast. Atmospheric and Climate Sciences **7**, 314–322 (2017)