International Journal of Forecasting 39 (2023) 992–1004

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

Deep learning models for visibility forecasting using climatological data

Luz C. Ortega, Luis Daniel Otero *, Mitchell Solomon, Carlos E. Otero, Aldo Fabregas

Department of Computer Engineering and Sciences, Florida Institute of Technology, Melbourne, FL 32901, USA

ARTICLE INFO

Keywords:
Weather forecasting
Neural network
Visibility forecast
Time series
Fog forecasting

ABSTRACT

Low visibility conditions affect safety and traffic operations, leading to adverse scenarios that often result in serious accidents. Due to the complexity and variability associated with modeling weather variables, visibility forecasting remains a highly challenging task and a matter of significant interest for transportation agencies nationwide. Given that the literature on single-step visibility forecasting is very scarce, this study explores the use of deep learning models for single-step visibility forecasting using time series climatological data. Five different deep learning models were developed, trained, and tested using data from two weather stations located in the US state of Florida, which is one of the top states nationwide dealing with low visibility problems. The authors provide discussions of the models' results and areas for future research.

1. Introduction

Low visibility conditions due to weather variables provide highly unsafe scenarios on roads, causing accidents and jeopardizing the operation, performance, and safety of transportation systems. Recent statistics from the US Federal Highway Administration show that there are over 31,500 yearly crashes nationwide due to low visibility, resulting in over 11,500 injuries and 500 fatalities (U.S. Department of Transportation. Federal Highway Administration. Road Weather Management, 2018). State-level statistics also provide a clear picture of the dangerous scenarios caused by low visibility on roads. A recent study concluded that between 2012 and 2016, the state of Texas led the nation in fatal crashes due to low visibility issues with 227, followed by 139 in California, and a similar amount in Florida (Lee & Abdel-Aty, 2019). These alarming statistics make low visibility issues a critical matter of concern for transportation agencies at both federal and state levels.

Over the years, various studies have been conducted to improve the effectiveness of visibility systems to reduce accidents on roads and highways. These systems are typically composed of data acquisition, data communication, data analysis, and data dissemination components. Within the data analysis component, fog detection and estimation of visibility distances have been identified as key areas that need further development (e.g., Abdel-Aty et al., 2015; Ray, 2016).

The type of visibility most pertinent to ground-level scenarios is horizontal visibility at the Earth's surface, which is defined by the International Civil Aviation Organization (ICAO) as "the greatest distance at which a black object of suitable dimensions, situated near the ground, can be seen and recognized when observed against a bright background" (ICAO, 2004). These visibility measurements are estimated by either forward scatter sensors or transmissometers at the site of Automated Surface Observing System (ASOS) station on an hourly basis.

Recent technological advances related to more efficiently collecting, storing, processing, and disseminating huge amounts of data have resulted in an increased number of machine learning algorithms developed for various

* Corresponding author.
E-mail address: lotero@fit.edu (L.D. Otero).

https://doi.org/10.1016/j.ijforecast.2022.03.009

applications. These advances provide opportunities to explore the use of machine learning approaches to address the visibility distance forecasting problem, which is a nonlinear complex problem characterized by temporal and spatial variability. Within the field of machine learning, deep learning models (i.e., neural networks with more than one hidden layer) provide a powerful and effective approach for solving such nonlinear complex problems.

Neural networks offer promise for time series data forecasting due to two main characteristics. First, neural networks are data-driven and self-adaptive in nature. Data-driven models are based on the analysis of the data characterizing a problem under study without explicit knowledge of the physical behavior of the system being modeled (Solomatine & Ostfeld, 2008). Moreover, data-driven models are adaptively developed based on the features presented from the data and not based on a priori assumptions about the statistical distribution of the data. The other key characteristics of neural networks relate to their capability of handling nonlinear data dependencies. When dealing with these types of complex problems, practitioners consider deep learning models to be a more practical approach than traditional linear approaches such as autoregressive integrated moving average (ARIMA) models (Khashei & Bijari, 2010).

Typically, the visibility forecasting problem involves developing models capable of effectively processing time series weather and/or video data to produce an output that characterizes visibility for a specified future time. This forecasting problem can be defined as a regression or classification problem. Regression problems deal with predicting a numerical output value such as visibility distance in miles, whereas classification problems deal with predicting a category or a class such as poor, moderate, or good visibility. The literature for visibility forecasting defined as a regression problem is very limited (Ismail Fawaz et al., 2019; Ortega et al., 2019). Most of the related studies found in the visibility forecasting literature treat the problem as a classification one.

The visibility forecasting task as a regression problem using time series data from ground weather stations has not been fully addressed in the academic literature. For this reason and the recent success of deep learning models for multivariate time series classification (e.g., Ismail Fawaz et al., 2019; Lecun et al., 2015; Wang et al., 2017), the authors were motivated to explore the use of deep learning models for visibility distance forecasting defined as a regression problem using data from Automated Surface Observing Systems (ASOS) stations. More specifically, this paper explores the following five models: multilayer perceptron (MLP), traditional convolutional neural network (TCNN), fully convolutional neural network (FCNN), multi-input convolutional neural network (MICNN), and long short-term memory (LSTM) network. These architectures were chosen because they are widely adopted in practice and academia for end-to-end nonlinear complex problems and have proven to be capable of learning from time series data.

This paper is organized into five sections. Section 1 introduces the visibility forecasting problem and its impact and significance to transportation systems. Section 2 reviews related work. Section 3 focuses on the methodology, description of the deep learning models, data description, and data analysis. Section 4 presents experimental results and discussion. Section 5 concludes with contributions to the literature and recommendations for future research.

1.1. Related work

Due to the dynamic nature of environmental parameters such as fog events, as well as the relatively small spatial scale on which these parameters tend to occur, the ability to produce accurate visibility estimations for transportation systems remains a complex challenge. The approaches presented in the academic literature involve two main types of data sources: sensor-based and camera-based. Sensor-based approaches use a suite of weather sensors to measure, collect, and disseminate weather data, including visibility distance. Typically, weather stations include sensors that provide data related to cloud and ceiling, visibility, precipitation, pressure, temperature (e.g., ambient, dew point, and dry-bulb temperatures), and humidity. The second approach uses data from surveillance and/or closed-circuit television (CCTV) camera systems to measure visibility. These two groups can be further divided into more categories based on solution approach (e.g., traditional statistical techniques and data-driven methodologies such as machine learning) and/or type of estimation (i.e., prediction or forecasting of quantitative or categorical values). The following subsections organize findings from the literature effort in terms of the type of data source.

1.1.1. Camera-based approaches

For transportation systems, camera-based research has mainly focused on road conditions, visibility distance measurement, and fog detection. Visibility distance estimation techniques using data from camera imagery systems are widely applied due to financial aspects and ease of installation of such systems. The main challenge with camera-based visibility estimation is how to restore the depth information that is lost from the original three-dimensional scene (Chaabani et al., 2017). A solution for this problem involves applying thresholding and segmentation techniques to search for points or regions of interest in an image to measure visibility based on luminance attenuation equations (Otero et al., 2018). However, this approach requires accurate geometric calibration of the camera and relies on the presence of reference objects with high contrasts in the scene.

The literature presents various studies that employed neural networks for visibility estimation. In Chaabani et al. (2018), the authors proposed a neural network approach for estimating short visibility distance ranges using a camera. The model was defined as a classification problem using a single hidden layer with 12 neurons and an output layer of six nodes, which denoted different visibility ranges. The model achieved an overall classification rate of 90.2%. However, the data used to train and test the models were synthetic images; that is, real-world data were not evaluated. In Li et al. (2017), the authors

proposed the use of a widely known pre-trained CNN (Krizhevsky et al., 2012) and a generalized regression neural network to estimate visibility distances from webcam weather images. The model was evaluated in a database with a visibility range between 5 km and 35 km, obtaining a predicting error of 3 km. Similarly, in You et al. (2019), the authors used outdoor images collected from the Internet to estimate visibility distances. The architecture of the model was based on a CNN and a recurrent neural network (RNN). The model achieved 90.3% accuracy for a visibility evaluation range of 300 to 800 m. The main drawbacks of this method were that the images used for evaluation were manually annotated, and the model was computationally costly. Another recent study proposed an approach based on deep integrated CNN for estimating distances from camera imagery (Palvanov & Cho, 2019). The model was defined as a classification problem and was evaluated using three image datasets, each with a different number of classes. In terms of classification accuracy, the model achieved 94% and 91.3% for short-range and long-range images, respectively. The authors calculated the mean square error; however, it is not clear how they obtained predicted visibility values, as the model outputs corresponded to probability estimates of each class membership and not a visibility value. Moreover, no estimation of future events was performed.

Image and video data availability are growing fast; however, there are still general limitations regarding visibility distance estimation approaches using such data. Most of the proposed models can be used only for daytime images, as images taken at night may not have enough features or good enough quality. Thus, night images require different image processing techniques. Model training of some of the approaches is highly time-consuming and computationally expensive. Some approaches focus only on the classification task and do not provide quantitative results.

1.1.2. Sensor-based approaches

Sensor-based approaches rely mainly on meteorological data (e.g., air temperature, wind direction, wind speed, and humidity)—mostly from weather and airport stations—to estimate visibility distances. The implementation of these approaches in new locations are likely to be more expensive than camera-based systems due to additional required hardware components.

Various studies have proposed the use of decision trees for fog forecasting using weather data (e.g., Dutta & Chaudhuri, 2015). Decision trees are a graphical representation that follows decision paths based on a set of rules. Despite the popularity of this approach in other fields, its application for visibility forecasting remains a challenge. A key issue involves dealing with the complexity of weather parameters such as fog to establish effective rules. Furthermore, these studies were defined as classification problems and only provided a categorical result (e.g., fog or no fog). Fuzzy logic approaches—designed to deal with complex, imprecise variables—have also been tested to forecast visibility (e.g., Murtha, 1995). However, the definition of membership functions and rules required to develop effective fuzzy logic models are also highly complex and subjective.

Although neural network models have been widely used in the atmospheric science field with promising success, their application for visibility forecasting is still limited (Gultepe et al., 2007). Various researchers have developed neural network models defined as classification problems in the areas of ceiling and visibility forecasting, achieving promising results (e.g., Bremnes & Michaelides, 2007; Marzban et al., 2007). However, these models provided only a visibility range or a visibility scenario such as low versus good visibility or fog versus no fog.

In Ortega et al. (2019), the authors used machine learning algorithms based on historical weather data to classify visibility into three classes: low visibility, moderate visibility, and good visibility. Despite achieving promising results, no quantitative results were provided, and the temporal component was not included. In Zhu et al. (2017), a deep learning model using weather data from an airport was implemented for visibility forecasting. Here, the authors reported an average RMSE of 0.443 miles, 0.452 miles, and 0.529 miles for lag cases of 3 h, 6 h, and 9 h, respectively, with an evaluation range from 0 to 5 km. The authors did not provide details about the model architecture; in fact, the only information provided was the type of neural network employed. Furthermore, the study indicated that the training and testing samples were randomly split, which is not considered the right technique when dealing with time series data forecasting (more information about this is provided in Section 4). In Wang et al. (2009), a three-layer (i.e., model with a single hidden layer) feed-forward risk neural network was proposed for visibility forecasting and set up as a classification model. The model used average daily values of weather variables, pollutant concentrations values, and a visibility value at a certain time to forecast the next day's visibility value at that time. The evaluation range was 0 to 6.21 mi, with a resulting average error of 1.3 mi.

The visibility estimation problem has also been addressed by using numerical weather prediction models (NWP) (Gultepe et al., 2006; Müller et al., 2007; Shi et al., 2012; Teixeira & Miranda, 2001). These models rely on physics and differential equations to produce visibility forecasts based on current weather data. Some of the factors that affect their accuracy include the density and quality of the observations used as inputs (i.e., initial conditions). Since numerical models are very sensitive to initial conditions, they do not achieve good results if the first weather observations used as inputs are not accurate or do not accurately represent current conditions, which is a common challenge for visibility estimation. In some cases, statistical and machine learning techniques can be used to calibrate for poor initial conditions in NPW models. Chmielecki and Raftery use Bayesian model averaging to calibrate the forecast and provide uncertainty estimations to an 8-model ensemble for visibility prediction in three different approaches (Chmielecki & Raftery, 2011). Herman and Schumacher develop a final forecast system consisting of a weighted average of four traditional machine learning classifiers (K-nearest neighbors, gradient boosting, random forest, and support vector machine) through model down-selection procedures, showing the value of statistical reforecasting of new model implementations (Herman & Schumacher, 2016).

Approaches using satellite data have also been used for visibility estimation (DeMott, 2007; Elrod & Saykally, 1995; Guidard & Tzanos, 2007). The main limitation of these models is the difficulty to differentiate between a low-lying cloud and fog using satellite data. Thus, these models are mostly useful to determine levels of visibility at mid and high altitudes and not at ground level; therefore, they are not appropriate for road visibility estimation.

Overall, studies have demonstrated that neural networks are capable of producing better visibility estimations than traditional statistical tools (e.g., linear and logistic regression) (Marzban et al., 2007), and given that the available literature addressing the visibility forecasting problem as a regression problem is scarce, the models presented in this paper provide a significant contribution to the academic literature.

2. Methodology

The solution approach employed in this paper for the visibility forecasting problem is shown in Fig. 1. The first step involved collecting time series weather data from two ASOS stations. The second step involved preprocessing and transforming the data into an appropriate input–output format for model development. The third step was to develop, train, and test neural network models corresponding to the following architectures: MLP, CNN, FCNN, MICNN, and LSTM. The final step was to evaluate the models for their capability and accuracy to forecast visibility outputs. The models were developed and implemented using Python and Keras, which is an open-source neural network library written in Python for model training and evaluation.

The forecasting problem was set up as a regression machine learning task, where the output response is a "visibility value" defined as the maximum horizontal single-step visibility (i.e., estimation of a visibility distance value for the next hour). Given that visibility values may change drastically in short periods of time, it is not necessary to capture long temporal patterns of historical data as input, but rather a few consecutive points. Three cases were considered for the number of lag observations provided as input: time series for 3 h, 6 h, and 9 h.

2.1. Neural network architectures

2.1.1. Multilayer perceptron

Despite being considered the simplest and most traditional architecture for deep learning models, MLP has achieved successful results in classification an regression tasks, including problems with time series data (Ahmed et al., 2010; Ismail Fawaz et al., 2019; Makridakis et al., 2018). MLP refers to a neural network architecture composed of multiple layers—an input layer, one or more hidden layers, and one output layer. Each layer is composed of units called nodes or neurons. MLP architectures are also known as fully connected (FC) networks since the nodes in layer l_i are connected to every other node in layer l_{i-1}. In addition, the nodes are organized in layers, with connections feeding forward from one layer to the next.

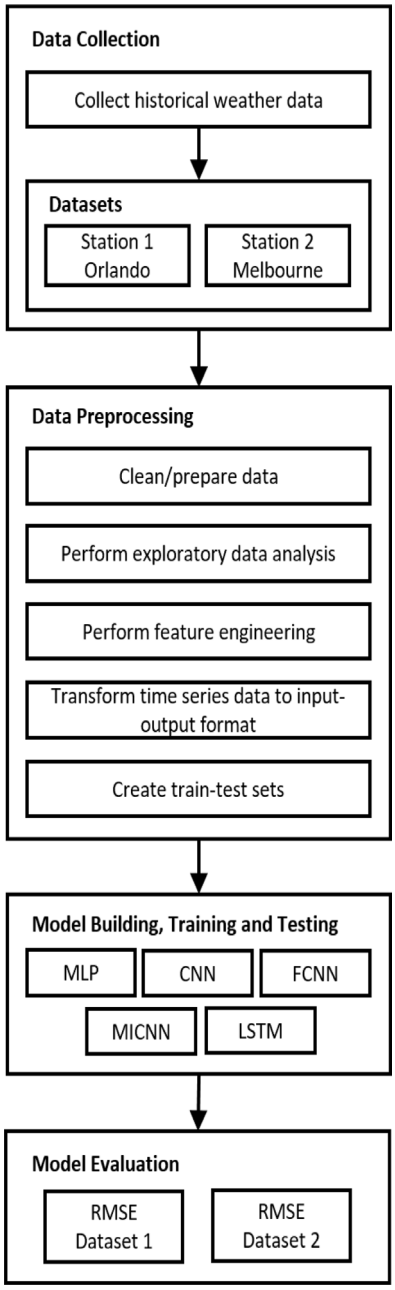

Fig. 1. Proposed solution approach.

In this paper, an MLP model is selected as the baseline model. The explored MLP architecture was inspired by a previously published model for time series classification (Wang et al., 2017). The architecture of this network consists of three FC layers. The first layer has 100 nodes and the other two have 50 nodes each. All three layers contain rectified linear unit (ReLU) activation functions. The final layer has one node with a linear activation function. The ReLU activation functions fulfill the nonlinearity characteristics of the dataset and prevent saturation of the gradient, thus helping the model to generalize well

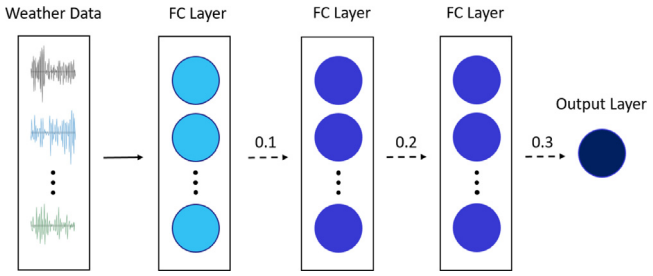

Fig. 2. Architectural overview of the explored MLP model (dash lines indicate the operation of dropout).

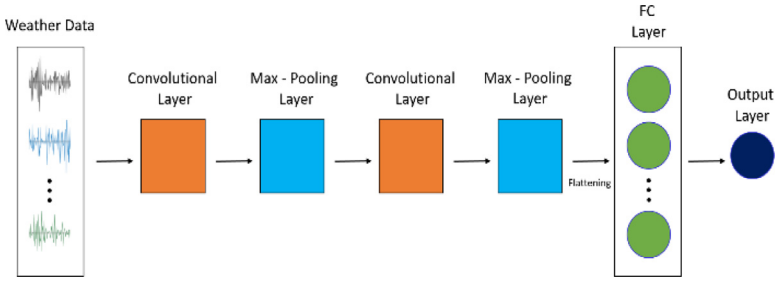

Fig. 3. Architectural overview of the explored TCNN model.

especially on small datasets (Nair & Hinton, 2010). Except for the output layer, each layer is followed by a dropout operation as a form of regularization to prevent overfitting and improve the generalization capacity (Srivastava et al., 2014). The dropout rates are {0.1, 0.2, 0.3}, respectively. Every model was trained for 500 epochs with the Adam optimization algorithm. The architecture of the MLP model is shown in Fig. 2.

2.1.2. Traditional convolutional neural network

CNNs are a type of feed-forward neural networks initially developed for classification tasks mostly involving image data or computer vision applications (e.g., facial recognition). Due to their effective capability of automatically extracting and learning features from a sequence of data, there is an increasing trend to implement these models using one-dimensional (1D) data such as text and time series. When implemented with 1D data, CNNs read across a sequence of previous observations and learn to extract features that are relevant for making predictions.

There are several variations of CNN architectures. Typically, the traditional architecture is a sequence of layers consisting of stacked convolutional and pooling layers followed by one or more FC layers that output the prediction of the model (Krizhevsky et al., 2012). The basis of CNNs is the convolution operation, which is a linear operation that involves the application of a filter to an input (i.e., the multiplication of a set of weights with the inputs). A convolution can be described as applying and sliding a filter over the input data. The filters, also known as kernels, are the equivalent of nodes in a regular neural network layer with learnable weights and biases. However, in a convolutional layer, the nodes are arranged in such a way that the nodes of each layer are only connected to a small region of the layer before it, instead of being connected to all the nodes. This local connectivity is achieved by replacing the weighted sums from the neural network with convolutions. The traditional CNN (TCNN) that is explored in this paper is based on a time series-specific version of a successful CNN model for image classification (Guennec et al., 2016). This model is composed of two convolutional pooling layers followed by an FC layer with a linear activation function. An overview of the architecture of this model is shown in Fig. 3. For both 1D convolutions, the ReLU activation function is used with 64 filters of size 3.

2.1.3. Fully convolutional neural network

The FCNN model architecture that is explored in this paper is derived from the TCNN model and it is composed of two 1D convolutional blocks with ReLU activation functions and 64 filters of size 3. It is important to mention that these blocks do not contain any local pooling layers. The result of the second convolutional block is averaged over the whole-time dimension by a global average pooling (GAP), as shown in Fig. 4. The objective of excluding any pooling operation after the convolution is to prevent overfitting. After the GAP layer, there is a fully connected layer with a linear activation function.

2.1.4. Multi-input convolutional neural network

The MICNN architecture explored in this paper is similar to the proposed TCNN one, with the difference being that the convolutional pooling blocks are applied independently on each of the time series dataset inputs. Each input series is handled by a separate CNN, and the output of each of these sub-models is combined before a prediction is made. In the MICNN architecture, the convolutions are applied in parallel to each time series, and each input has its own filters. This architecture has two 1D convolutional pooling layers for each input of the dataset before

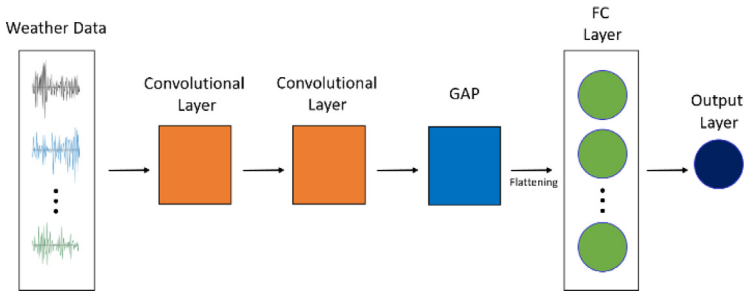

Fig. 4. Architectural overview of the explored FCNN model.

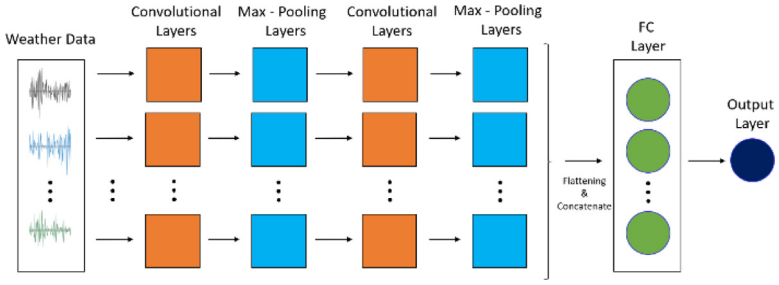

Fig. 5. Architectural overview of the explored MICNN model.

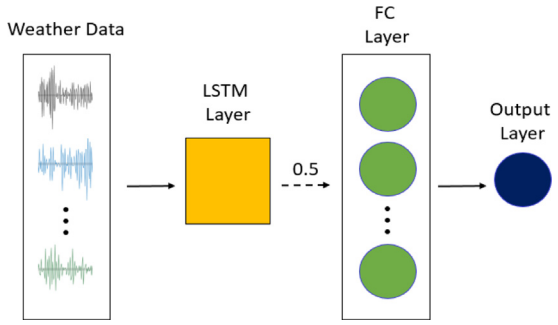

Fig. 6. Architectural overview of the explored LSTM model.

the FC layer, as shown in Fig. 5. The parameters for each input CNN model are the same as the ones used for the proposed TCNN.

2.1.5. Long short-term memory network

The LSTM network architecture is a type of RNN which is able to learn across a sequence of data (Hochreiter & Schmidhuber, 1997). The main difference between an LSTM and a feed-forward network is that the former reads one-time step of the sequence at a time and builds up an internal state representation used as a learned context for making a prediction, while the latter reads across the entire input vector. At the end of the sequence, each node of a hidden LSTM layer outputs a vector of values that represent what the layer learned from the input sequence. This vector is interpreted by a FC layer followed by the output layer. Although developed for sequenced data, LSTMs have not proven effective on time series forecasting problems where the output is a function of recent observations (e.g., visibility distance) (Makridakis et al., 2018). The explored architecture is shown in Fig. 6. The LSTM layer has 100 nodes, followed by a dropout operation of rate {0.5} and a FC layer of 50 nodes before a single node output layer.

2.2. Data

The data used for this study corresponded to local climatological data (LCD) from two ASOS stations located in the state of Florida, specifically in the cities of Melbourne and Orlando. LCD datasets from both stations were retrieved from the National Oceanic and Atmospheric Administration (NOAA) website. Climatological studies have shown that January typically is the peak month for heavy fog (i.e., horizontal visibility of less than one-quarter of a mile) in most of the Southeastern region of the USA (Hardwick, 1973; PEACE, 1969). In the state of Florida, fog-related studies indicate that low visibility conditions usually occur between the months of December and March (Ray & Lefran, 2015). Therefore, the data used to develop, train, and test the models for this research included hourly weather observations from December 1, 2018 to March 31, 2019.

The variables used to develop the models consisted of one output and seven input parameters. The output variable was visibility, which in the dataset was defined as the horizontal distance in miles that an object could be seen and identified. The input variables for the models were selected based on domain knowledge, studies from the academic literature, and data availability. Correlation analyses between the variables were conducted using the Spearman rank correlation coefficient for both datasets. As with any bivariate correlation analysis, the

Spearman correlation measures the strength of association between two variables in a value between −1 and +1, where a value of 1 is total positive linear correlation, 0 indicates no linear correlation (a nonlinear relationship might exist), and −1 is total negative correlation. Scatterplots were examined to determine possible relationships between variables with correlations coefficients close to zero. The following subsections describe the selected input variables for the models.

2.2.1. Difference between dry-bulb temperature and dew point temperature

Low visibility conditions caused by fog usually form when the difference between the air temperature and dew point is less than 4.5° F. Initial correlation results showed a weak linear relationship between the temperature variables (dry bulb and dew point) and visibility. However, the literature related to visibility indicates the opposite. A new feature was created by calculating the difference between the "dry-bulb temperature" and the "dew point temperature". The new feature had a stronger linear correlation with visibility than the two temperature variables used for its creation.

2.2.2. Wet-bulb temperature

This variable refers to the adiabatic saturation temperature and it is an indicator of moisture in the air, an important factor for visibility conditions (American Meteorological Society, 2019). The value of wet-bulb temperature is between the dew point and the dry-bulb temperature. Wet-bulb temperature is higher than the dew point temperature and lower than the dry-bulb temperature, except at saturation (100% relative humidity), when the three temperatures are identical.

2.2.3. Relative humidity

The relative humidity is the ratio of the amount of water vapor in the air compared to the amount required for saturation (Haynes et al., 2016). When the air is 100% saturated (i.e., when the relative humidity is at its maximum value), the conditions are ideal for low visibility (Gliessman, 2007), especially related to fog formation (Gultepe et al., 2019).

2.2.4. Wind direction

Wind direction data points represent the angle, measured in a clockwise direction, between true north and the direction from which the wind is blowing. Meteorological experts consider wind direction as an indicator for low visibility scenarios, especially when there is a nearby source of moisture or heat (Cools et al., 2010; Fabbian et al., 2007; Gultepe et al., 2007).

2.2.5. Wind speed

This variable represents the wind speed at the time of observation given in miles per hour (mph). Wind speed plays an important factor in fog formation and dissipation. High wind speeds may dissipate mist before it forms into fog, and low wind speeds allow for turbulent mixing, spreading cooling vertically and deepening fog layers (Nair & Hinton, 2010).

Table 1
Hyperparameter sets.

Parameter	Architecture		
	MLP	TCNN	LSTM
Activation function	{relu, sigmoid}	{relu, sigmoid}	{relu, sigmoid}
Nodes per layers	{50,100}	{16,64}	{50,100}
Optimizer	{adam, sgd}	{adam, sgd}	{adam, sgd}

2.2.6. Sky conditions

Typically, low visibility conditions occur under clear, nocturnal skies. Sky conditions based on sky coverage are considered an important determinant of visibility conditions due to radiative fog formation (Cools et al., 2010; Gultepe et al., 2007).

2.2.7. Time of the day

Low visibility conditions typically occur during early morning hours and late night. Subject matter experts and researchers in the academic literature indicate that time of the day is a relevant determinant of low visibility conditions (Bremnes & Michaelides, 2007; Cools et al., 2010; Gultepe et al., 2007; Marzban et al., 2007; Ortega et al., 2019).

3. Experimental results

3.1. Experimental settings

Finding optimal configurations for deep learning models is a time-consuming task. An often-used approach deals with the use of hyperparameters, which are parameters that are not directly determined through model training, but instead are chosen before the training phase. Hyperparameters may include the structure of the network (e.g., number of nodes or hidden layers), training algorithm, and time steps, among others. In this paper, the literature related to deep learning for time series (e.g., model architectures and hyperparameters that have achieved successful results for time series classification), paired with empirically driven hyperparameter choices, was used for hyperparameter tuning.

Hyperparameter tuning was performed for the MLP, TCNN, and LSTM models using the Orlando data and split 3 as shown in Fig. 7. Hyperparameter tuning was not performed for the FCNN and MICNN models because these models are derived from the TCNN; thus, the same hyperparameters were used. Furthermore, the same hyperparameters were used to train and test the Melbourne dataset to investigate if the models could perform better with this dataset compared to the Orlando one. Table 1 shows the hyperparameter sets for each model.

3.2. Evaluation settings

Classical machine learning cross-validation techniques, such as k-fold cross-validation and hold-out methods, are very effective in evaluating model performance, assuming that there is no correlation between observations. The assumption of independent and identically distributed observations does not hold well with time series data

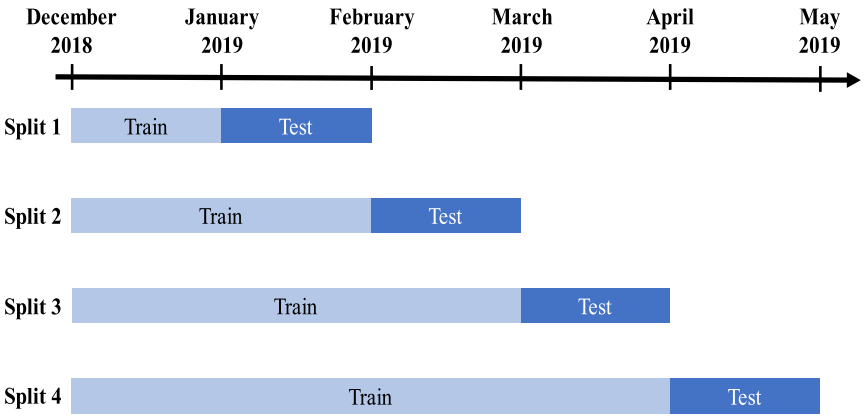

Fig. 7. Four-fold walk forward sequence validation.

because there is an implicit dependence on previous observations due to the temporal component. When modeling using time series data, the ordering imposed by the time index is important to capture temporal relationships, if any, between the variables. Thus, the randomization in classical cross-validation techniques does not preserve the time ordering, resulting in unreasonable correlations between the training and testing sets on time series data. Since time series data are often strongly correlated along the time axis, the randomization will likely have the effect that for each observation in the test set, several strongly correlated observations exist in the train set. This problem potentially leads to inflated performance metrics on the test set, yielding incorrect values for model performance. A frequently proposed validation alternative is to use a single train-test split based on time, where the dataset is split at an arbitrary point. Here, the first portion of the data is used for training and the second one for testing. This approach is useful when both the training and testing sets are representative of the original problem. However, this approach could result in an implicit bias toward the most recent period.

To overcome the issues with time series weather data, the authors evaluated the models using an intuitive walk forward sequence validation. With this approach, the forecasting models were periodically re-trained, incorporating data available at different points in time. The performance of the models was based on the average test error of the number of times the model was trained. This approach accounts for the models' performance at different time windows. For this research, a four-fold walk forward sequence validation was performed, where the first training set consisted of observations from the month of December, while the testing set consists of observations from the month of January. The following iteration used all the data from the previous iteration for the train set, wherein the test set consisted of the month immediately following the training set.

Fig. 7 illustrates the proposed approach for model validation. This figure shows a time progression line corresponding to the four train-test splits, clearly specifying the monthly data used for training and testing. This type of model evaluation method more accurately describes weather variables at prediction time since the model is based on past data and predicts future data (Gultepe et al., 2007).

3.3. Numerical results

The performance metric used to measure the forecasting accuracy of the proposed models was the root mean squared error (RMSE). This metric is the square root of the average of squared differences between the predicted and actual values (see Eq. (1)), which were to be minimized. The RMSE, which has the same units as the quantity being estimated (i.e., miles), penalizes predicted values significantly more as their distance from actual values increases (i.e., significantly penalizes inaccuracy in the forecasted values). The RMSE is calculated as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2} \tag{1}$$

where n is the size of the test dataset, y_j represents the actual value, and $\hat{y}_j$ represents the forecasted value.

Table 2 shows the performance metrics achieved by the five deep learnings (top) and two traditional models (bottom) on each of four train-test splits with the three different time lags at two weather stations. Fig. 8 shows visibility predicted by the MLP corresponding to split 3 and 6 h of input data at the Orlando weather station.

In general, dataset and split seem to be larger influences on performance compared to model type or lag time. The best 10 RMSE values occur on Orlando dataset split 3, with 9 of these being from deep learning models, while 6 of the 10 best models on the Melbourne dataset are either LR or ARIMA on split 4. Conversely, the worst 10 performances on both datasets are on split 2 with only 3 of these 20 being a traditional model (ARIMA on the Orlando dataset). The proposed deep learning architectures achieved better results on the Orlando dataset which likely stems from the fact that hyperparameter tuning of these models was performed on split 3 of this dataset. To further investigate possible influences on model performance, the analysis was extended by way of ranking comparisons in the following section.

Table 2
RMSE (miles) by the seven explored models.

Model	Input data (h)	Orlando Train-test splits				Melbourne Train-test splits			
		Split 1	Split 2	Split 3	Split 4	Split 1	Split 2	Split 3	Split 4
MLP	3	1.215	1.439	0.839	1.132	1.865	2.275	1.546	0.907
	6	1.104	1.493	0.914	1.139	1.793	2.166	1.500	0.948
	9	1.170	1.586	0.882	1.085	1.732	2.253	1.618	0.941
TCNN	3	1.203	1.427	0.832	1.171	1.900	2.357	1.657	0.938
	6	1.222	1.641	0.958	1.155	2.052	2.381	1.816	1.064
	9	1.136	1.442	0.884	1.178	1.916	2.245	1.517	0.955
FCNN	3	1.184	1.546	0.831	1.153	1.963	2.260	1.563	0.934
	6	1.251	1.400	0.871	1.18	1.847	2.173	1.553	0.869
	9	1.175	1.611	0.950	1.17	2.095	2.454	1.733	1.102
MICNN	3	1.058	1.466	0.857	1.218	1.905	2.117	1.520	0.898
	6	1.269	1.449	0.851	1.314	2.111	2.469	1.874	0.991
	9	1.201	1.437	0.856	1.206	1.729	2.036	1.566	0.849
LSTM	3	1.183	1.474	0.800	1.131	1.818	2.185	1.492	0.858
	6	1.181	1.470	0.912	1.148	1.872	2.097	1.594	0.859
	9	1.247	1.462	0.933	1.142	1.864	2.283	1.701	0.981
LR	3	1.112	1.365	0.941	0.910	1.599	1.942	1.377	0.796
	6	1.119	1.368	0.950	0.910	1.601	1.946	1.372	0.789
	9	1.121	1.366	0.953	0.910	1.608	1.947	1.370	0.788
ARIMA	3	1.135	1.530	0.884	0.988	1.865	1.981	1.433	0.824
	6	1.128	1.530	0.903	0.984	1.951	1.892	1.463	0.767
	9	1.118	1.518	0.900	0.985	1.881	1.893	1.443	0.834

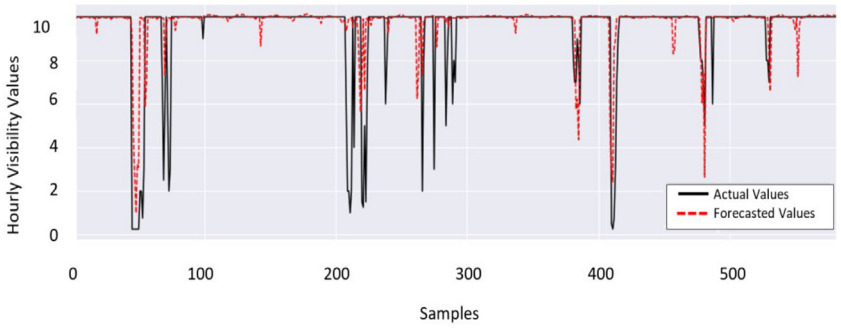

Fig. 8. Results with MLP architecture and 6 h input data (Orlando).

3.4. Ranking comparison

The rankings of the seven models were examined to determine their relative performance by using a multiple comparison with the best (MCB) procedure (Koning et al., 2005; Makridakis et al., 2018). For this nonparametric test, there are K treatments ($k = 1, 2, \ldots, K$), each having N observations ($n = 1, 2, \ldots, N$) to forecast the next time step ($t + 1$) using P different lag cases ($p = 1, 2, \ldots, P$), ranked from 1 to K (with 1 being the best and K being the worst) based on the RMSE. Since the objective was to compare forecasting models, $K = 7$ models were chosen as treatments. For each model k, there is an average rank $\bar{R}_k$ with $\alpha = 95\%$ confidence limit, defined in Eq. (2):

$$\bar{R}_k \pm 0.5 q_{\alpha K} \sqrt{\frac{K(K+1)}{12N}} \tag{2}$$

where $q_{\alpha K}$ is the upper α percentile point of the range of K independent standard normal variables. The results of the test can be interpreted as follows: the model with the lowest average rank becomes the best model, and its upper boundary becomes the reference value. Thus, all models with confidence intervals entirely above the reference value without any overlap perform significantly worse than the best model. An overall comparison is first visualized by assuming that test data comes from different datasets, splits come from the same distribution, and the lag parameter has no significant effect on model RMSE (Fig. 9).

Fig. 9 provides a visualization of the multiple comparison with the best (MCB) analysis. Here, any model whose error bars overlap with the critical difference threshold (dotted grey line) is said to perform equivalent to the best model with statistical significance. The figure indicates that the FCNN, MICNN, and TCNN models are significantly worse than the others for the forecasting task. The assumptions of the overall MCB analysis are tested via three-way analysis of variance (ANOVA) by modeling the RMSE as a function of datasets, train-test splits, lag cases, and their interactions (Table 3).

The ANOVA results indicate that lag cases and interactions including lag cases do not significantly impact RMSE. However, dataset, split, and their interaction

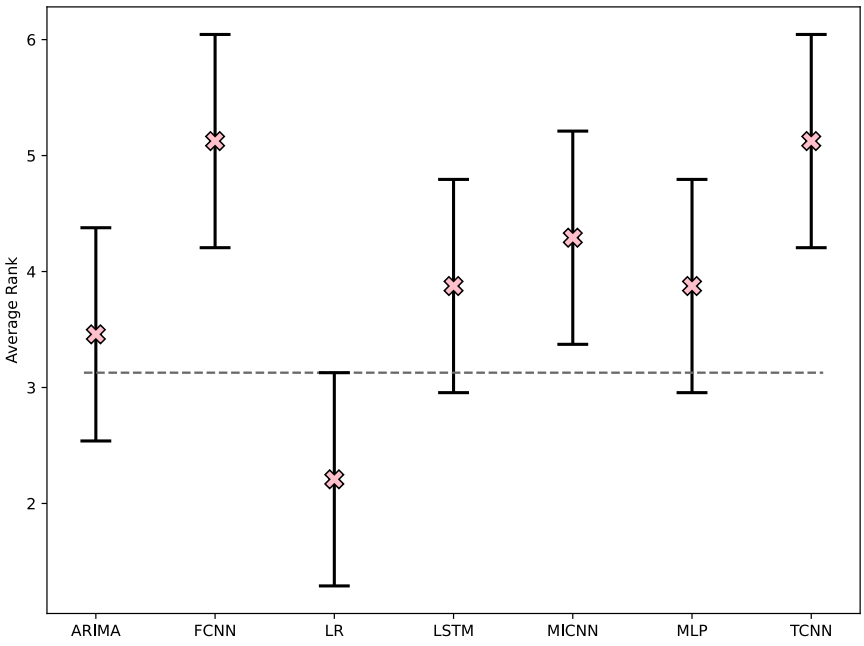

Fig. 9. Visualization of the multiple comparison with the best (MCB) analysis.

Table 3
ANOVA results on forecasted RMSE values modeled as a function of dataset, train-test split, and lag case.

	SSE	df	*F*	P(>F)	Significant?
Intercept	324.206495	1	20016.071	2.005E−156	–
Dataset	12.221833	1	754.559	4.037E−59	Yes
Split	15.797907	3	325.114	6.772E−64	Yes
Lag	0.030461	2	0.940	3.929E−01	No
Dataset x Split	2.454786	3	50.518	2.276E−22	Yes
Dataset x Lag	0.006506	2	0.201	8.183E−01	No
Split x Lag	0.021837	6	0.225	9.682E−01	No
Dataset x Split x Lag	0.003566	6	0.037	9.998E−01	No
Residual	2.332413	144	—	-	

effects are large enough to significantly impact RMSE. Therefore, it is concluded that the underlying data distributions of dataset and train-test split are significantly different, which may have given rise to the nonmonotonic relationship between amount of data used for training and test performance.

Following this result, the logical next step was to re-visualize the MCB analysis on unique dataset and train-test-split combinations, using lag cases as replicates, as shown in Fig. 10. Here, it was observed that the ranking comparison yields only one instance where any model's performance is significantly worse than the best, occurring for the MICNN model in the Melbourne dataset, split 4. It follows from this analysis that the explored deep learning methods are, overall, performing similar to traditional methods for visibility forecasting on the two Florida-based visibility prediction datasets.

4. Discussion

Applying deep learning methodologies to low-data situations is sometimes necessary for establishing baselines and data-dependent trends of model performance in comparison with traditional methods. It is expected for data-driven models to enjoy increased generalization performance as the amount of data used to train them is increased, but this is not always the case as observed in the RMSE result of Table 2. An increase in average RMSE from split 1 to split 2 was observed for all models in both datasets, as well as an increase in RMSE from split 3 to split 4 for deep learning models and ARIMA on the Orlando dataset. It would seem that split 2 presents a relatively harder problem than split 1, despite having more data to train on. This departure from the expected trend may be an indicator of the large variance in underlying data distribution between each dataset-split combination, which may be exacerbated by the walk-forward-in-time data splitting methodology being ordered in time rather than completely randomized. For example, the climatological conditions of the test month of split 2 may be much farther out of distribution from its training months than from split 3 or 4; thus, yielding the poorer generalization that was observed on split 2 across all models. The ANOVA analysis in Table 3 confirmed the apparent statistical significance of station location (dataset) and train-test split over lag case, allowing the lag cases to

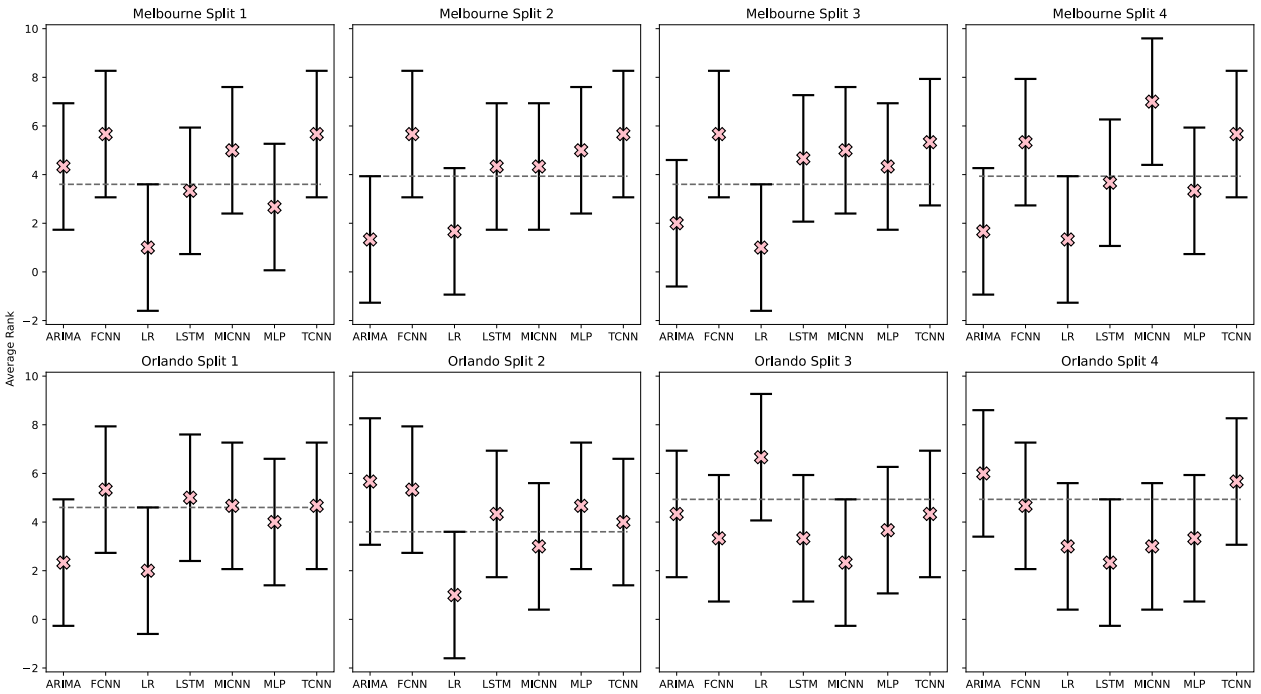

Fig. 10. MCB analysis on unique combinations of dataset and train-test split.

be treated as replicates in a partitioned version of the MCB analysis (Fig. 10), where the analysis is repeated over unique station locations and splits independently.

In the partitioned MCB analysis performed in this work, traditional methods are deemed the best in 6 out of 8 comparisons while deep learning methods take the remainder. However, both deep learning model wins occur from the Orlando dataset, on which it was also observed a steadily decreasing trend in average rank of deep learning models with increasing training set size, especially for the LSTM. As its name suggests, the LSTM has been shown to exhibit memory across time, making them especially suited for learning time-series data compared to other deep learning methods. Therefore, the walk-forward cross-validation method in time should be an effective method of illuminating the robustness of time-dependent feature learning. From this point of view, the LSTM is the deep learning model that will most likely reflect a performance boost as the training data increases and is recommended for future applications of this kind.

5. Conclusions

This study explored the use of deep learning models for visibility distance forecasting using time series data from ground weather stations in Florida. The authors explored five deep learning models, one with an MLP architecture, three different convolutional architectures (derived from a traditional CNN), an LSTM model, and two traditional models (LR and ARIMA). The recommendation to develop LSTM models seems adequate given the steady improvement of model rank with the size of training set and the proven ability of this type of architecture to automatically extract time-dependent features from raw input data.

The forecasting problem was set up as a regression machine learning task, where the output response is a "visibility value" defined as the maximum horizontal single-step visibility (i.e., estimation of a visibility distance value for the next hour). Given that visibility values may change drastically in short periods of time, it was not necessary to capture long temporal patterns of historical data as input, but rather a few consecutive points. Three cases were considered for the number of lag observations provided as input: time series data for the past 3 h, 6 h, and 9 h.

The climatological data used to develop, train, and test the models were collected from two ASOS stations located in the cities of Melbourne and Orlando, in the state of Florida. The data included hourly weather observations from December 1, 2018 to April 30, 2019. These months were identified in previous studies as having the most incidences of low visibility conditions due to fog in Florida.

Numerical accuracy results and analyses indicated a departure of expected performance trends for data-driven models, which are explained by a number of experimental design choices. Hyperparameter tuning was performed on split 3 of the Orlando dataset, giving rise to the superior performance of deep learning methods on this split and on the Orlando dataset by extension. This indicated that the performance of the models depends on the data used for the hyperparameter tuning, and therefore, on the location of the data. An ANOVA performed over station location, split, and lag case confirmed that performance is significantly different, which was attributed to apparent differences of underlying dataset distribution between station locations and train-test splits. Treating lag cases

as replicates, the MCB analysis was repeated over unique station locations and train-test splits, showing only one instance where a deep learning method was significantly worse than the best model. Therefore, it is concluded that the explored deep learning methods are competitive with established traditional techniques for visibility forecasting on the Melbourne and Orlando datasets. Due to its steadily decreasing average rank and theoretical considerations, it is recommended that the LSTM be explored for further application and that the expected performance increase would most likely be realized on a much larger dataset of this kind, which would ideally span many more years and station locations.

The application of deep learning models for visibility forecasting as a regression problem is very scarce in the academic literature. The analysis presented in this paper may encourage the visibility forecasting community to conduct further research in this very important topic area and also to collect more data to fully realize the capabilities of deep learning models. Further research could investigate other factors that might affect visibility such as landforms, pollutants coefficients, and water bodies. Incorporating these factors as input features may increase the forecasting performance of the models. In addition, it will be necessary to design and develop a software tool that facilitates the implementation of the proposed solution approach for practical purposes. Critical to this step will be a robust requirements engineering approach to ensure that critical functional needs are incorporated into the proposed solution (Ejnioui et al., 2012, 2013), including an evaluation of uncertainty in deep learning model predictions (Kuleshov et al., 2018; Loquercio et al., 2020).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abdel-Aty, M., Radwan, E., Oloufa, A., & Rodgers, M. (2015). A comprehensive investigation of visibility problems on highways: Developing real time monitoring and prediction system for reduced visibility and understanding traffic and human factors implications.

Ahmed, N. K., Atiya, A. F., Gayar, N. El., & El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Economic Review*, http://dx.doi.org/10.1080/07474938.2010.481556.

American Meteorological Society (2019). Wet bulb temperature. Glossary of meteorology [www document].

Bremnes, J. B., & Michaelides, S. C. (2007). Probabilistic visibility forecasting using neural networks. In *Fog and Boundary Layer Clouds: Fog Visibility and Forecasting* (pp. 1365–1381). Springer.

Chaabani, H., Kamoun, F., Bargaoui, H., Outay, F., & Yasar, A.-U.-H. (2017). A neural network approach to visibility range estimation under foggy weather conditions. *Procedia Computer Science*, *113*, 466–471. http://dx.doi.org/10.1016/J.PROCS.2017.08.304.

Chaabani, H., Werghi, N., Kamoun, F., Taha, B., Outay, F., & Yasar, A.-U.-H. (2018). Estimating meteorological visibility range under foggy weather conditions: A deep learning approach. *Procedia Computer Science*, *141*, 478–483. http://dx.doi.org/10.1016/J.PROCS.2018.10.139.

Chmielecki, R. M., & Raftery, A. E. (2011). Probabilistic visibility forecasting using Bayesian model averaging. *Monthly Weather Review*, *139*(5), 1626–1636.

Cools, M., Moons, E., & Wets, G. (2010). Assessing the impact of weather on traffic intensity. *Weather, Climate, and Society*, *2*, 60–68. http://dx.doi.org/10.1175/2009WCAS1014.1.

DeMott, P. J. (2007). Progress and issues in quantifying ice nucleation involving atmospheric aerosols. In *Nucleation and Atmospheric Aerosols* (pp. 405–417). Springer.

Dutta, D., & Chaudhuri, S. (2015). Nowcasting visibility during wintertime fog over the airport of a metropolis of India: decision tree algorithm and artificial neural network approach. *Natural Hazards*, *75*, 1349–1368. http://dx.doi.org/10.1007/s11069-014-1388-9.

Ejnioui, A., Otero, C. E., & Otero, L. D. (2012). A simulation-based fuzzy multi-attribute decision making for prioritizing software requirements. In *RIIT'12 - Proceedings of the ACM Research in Information Technology*. http://dx.doi.org/10.1145/2380790.2380800.

Ejnioui, A., Otero, C. E., & Otero, L. D. (2013). Prioritisation of software requirements using grey relational analysis. *International Journal of Computer Applications in Technology*, *47*, http://dx.doi.org/10.1504/IJCAT.2013.054344.

Elrod, M. J., & Saykally, R. J. (1995). Vibration–rotation–tunneling dynamics calculations for the four-dimensional (HCl) 2 system: A test of approximate models. *The Journal of Chemical Physics*, *103*, 921–932.

Fabbian, D., de Dear, R., Lellyett, S., Fabbian, D., de Dear, R., & Lellyett, S. (2007). Application of artificial neural network forecasts to predict fog at canberra international airport. *Weather Forecast*, *22*, 372–381. http://dx.doi.org/10.1175/WAF980.1.

Gliessman, S. R. (2007). *Agroecology: The ecology of sustainable food systems*. CRC Press.

Guennec, A. Le, Malinowski, S., & Tavenard, R. (2016). Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD Work. adv. anal. learn. temporal data.*

Guidard, V., & Tzanos, D. (2007). Analysis of fog probability from a combination of satellite and ground observation data. *Pure and Applied Geophysics*, *164*, 1207–1220.

Gultepe, I., Agelin-Chaab, M., Komar, J., Elfstrom, G., Boudala, F., & Zhou, B. (2019). A meteorological supersite for aviation and cold weather applications. *Pure and Applied Geophysics*, *176*(5), 1977–2015.

Gultepe, I., Müller, M. D., & Boybeyi, Z. (2006). A new visibility parameterization for warm-fog applications in numerical weather prediction models. *Journal of Applied Meteorology and Climatology*, *45*, 1469–1480.

Gultepe, I., Tardif, R., Michaelides, S. C., Cermak, J., Bott, A., Bendix, J., Müller, M. D., Pagowski, M., Hansen, B., Ellrod, G., Jacobs, W., Toth, G., & Cober, S. G. (2007). Fog research: A review of past achievements and future perspectives. *Pure and Applied Geophysics*, *164*, 1121–1159. http://dx.doi.org/10.1007/s00024-007-0211-x.

Hardwick, W. C. (1973). Monthly fog frequency in the continental United States. *Monthly Weather Review*, *101*, 763–766. http://dx.doi.org/10.1175/1520-0493(1973)101<:MFFITC>.3.CO;2.

Haynes, W. M., Lide, D., & Bruno, T. J. (2016). *CRC handbook of chemistry and physics: a ready-reference book of chemical and physical data* (97th ed.). Boca Raton, FL: CRC Press.

Herman, G. R., & Schumacher, R. S. (2016). Using reforecasts to improve forecasting of fog and visibility for aviation. *Weather and Forecasting*, *31*(2), 467–482.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, http://dx.doi.org/10.1162/neco.1997.9.8.1735.

ICAO, A. (2004). 3: Annex 3 to the convention on international civil aviation: Meteorological service for international air navigation.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, http://dx.doi.org/10.1007/s10618-019-00619-1.

Khashei, M., & Bijari, M. (2010). An artificial neural network (p, d, q) model for timeseries forecasting. *Expert Systems With Applications*, http://dx.doi.org/10.1016/j.eswa.2009.05.044.

Koning, A. J., Franses, P. H., Hibon, M., & Stekler, H. O. (2005). The M3 competition: Statistical tests of the results. *International Journal of Forecasting*, *21*(3), 397–409.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097–1105.

Kuleshov, V., Fenner, N., & Ermon, S. (2018). Accurate uncertainties for deep learning using calibrated regression. arXiv [cs.LG]. arXiv. http://arxiv.org/abs/1807.00263.

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444. http://dx.doi.org/10.1038/nature14539.

Lee, J., & Abdel-Aty, M. A. (2019). *Investigation of low visibility related crashes in florida. orlando*.

Li, S., Fu, H., & Lo, W.-L. (2017). Meteorological visibility evaluation on webcam weather image using deep learning features. *International Journal of Computer Theory and Engineering*, http://dx.doi.org/10.7763/ijcte.2017.v9.1186.

Loquercio, A., Segu, M., & Scaramuzza, D. (2020). A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, *5*(2), 3153–3160.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *PLoS One*, *13*, Article e0194889. http://dx.doi.org/10.1371/journal.pone.0194889.

Marzban, C., Leyton, S., & Colman, B. (2007). Ceiling and visibility forecasts via neural networks. *Weather Forecast*, *22*, 466–479.

Müller, M. D., Schmutz, C., & Parlow, E. (2007). A one-dimensional ensemble forecast and assimilation system for fog prediction. *Pure and Applied Geophysics*, *164*, 1241–1264.

Murtha, J. (1995). Applications of fuzzy logic in operational meteorology. (pp. 42–54). Sci. Serv. Prof. Dev. Newsletter, Can. Forces Weather Serv.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proc. 27th Int. conf. mach. learn.* doi: 10.1.1.165.6419.

Ortega, L., Otero, L. D., & Otero, C. (2019). Application of machine learning algorithms for visibility classification. In *2019 IEEE International systems conference* (pp. 1–5). http://dx.doi.org/10.1109/SYSCON.2019.8836910.

Otero, L., Moyou, M., Peter, A., & Otero, C. E. (2018). Towards a remote sensing system for railroad bridge inspections: a concrete crack detection component. In *SoutheastCon 2018* (pp. 1–4).

Palvanov, A., & Cho, Y. (2019). VisNet: Deep convolutional neural networks for forecasting atmospheric visibility. *Sensors*, *19*(1343), http://dx.doi.org/10.3390/s19061343.

PEACE, R. L. (1969). Heavy-fog regions in the conterminous United States. *Monthly Weather Review*, *97*, 116–123. http://dx.doi.org/10.1175/1520-0493(1969)097<0116:HRITCU>2.3.CO;2.

Ray, P. S. (2016). *Evaluation of fog predictions and detection, phase 2: draft final report*. Florida State University. Dept. of Earth, Ocean, and Atmospheric Science.

Ray, P. S., & Lefran, D. (2015). Evaluation of fog predictions and detection.

Shi, C., Wang, L., Zhang, H., Zhang, S., Deng, X., Li, Y., & Qiu, M. (2012). Fog simulations based on multi-model system: a feasibility study. *Pure and Applied Geophysics*, *169*, 941–960.

Solomatine, D. P., & Ostfeld, A. (2008). Data-driven modelling: some past experiences and new approaches. *Journal of Hydroinformatics*, http://dx.doi.org/10.2166/hydro.2008.015.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Teixeira, J., & Miranda, P. M. A. (2001). Fog prediction at lisbon airport using a one-dimensional boundary layer model. *Meteorological Applications*, *8*, 497–505.

U. S. Department of Transportation. Federal Highway Administration. Road Weather Management (2018). How do weather events impact roads? [www document].

Wang, Z., Yan, W., & Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. In *Proceedings of the international joint conference on neural networks*. http://dx.doi.org/10.1109/IJCNN.2017.7966039.

Wang, K., Zhao, H., Liu, A., & Bai, Z. (2009). The risk neural network based visibility forecast. In *2009 Fifth international conference on natural computation* (pp. 338–341). http://dx.doi.org/10.1109/ICNC.2009.152.

You, Y., Lu, C., Wang, W., & Tang, C. K. (2019). Relative CNN-rnn: Learning relative atmospheric visibility from images. *IEEE Transactions on Image Processing*, http://dx.doi.org/10.1109/TIP.2018.2857219.

Zhu, L., Zhu, G., Han, L., & Wang, N. (2017). The application of deep learning in airport visibility forecast. *Atmospheric and Climate Sciences*, http://dx.doi.org/10.4236/acs.2017.73023.