# A mixture of experts model for predicting persistent weather patterns

M. Pérez-Ortiz*, P.A. Gutiérrez†, P. Tino‡, C. Casanova-Mateo§ and S. Salcedo-Sanz¶
*Computer Laboratory, University of Cambridge (UK), email: mp867@cam.ac.uk
†Dpt. of Computer Science and Numerical Analysis, University of Córdoba (Spain)
‡School of Computer Science, University of Birmingham (UK)
§Department of Civil Engineering: Construction, Infrastructure and Transport, Universidad Politécnica de Madrid, Madrid (Spain)
¶Department of Signal Processing and Communications, University of Alcalá (Spain)

***Abstract*—Weather and atmospheric patterns are often persistent. The simplest weather forecasting method is the so-called persistence model, which assumes that the future state of a system will be similar (or equal) to the present state. Machine learning (ML) models are widely used in different weather forecasting applications, but they need to be compared to the persistence model to analyse whether they provide a competitive solution to the problem at hand. In this paper, we devise a new model for predicting low-visibility in airports using the concepts of mixture of experts. Visibility level is coded as two different ordered categorical variables: cloud height and runway visual height. The underlying system in this application is stagnant approximately in 90% of the cases, and standard ML models fail to improve on the performance of the persistence model. Because of this, instead of trying to simply beat the persistence model using ML, we use this persistence as a baseline and learn an ordinal neural network model that refines its results by focusing on learning weather fluctuations. The results show that the proposal outperforms persistence and other ordinal autoregressive models, especially for longer time horizon predictions and for the runway visual height variable.**

***Index Terms*—mixture of experts, persistence model, dynamic systems, ordinal classification, ordinal regression, autoregressive models, neural networks, low-visibility**

## I. Introduction

The persistence model, commonly used for weather forecasting, assumes that the conditions at the time of the forecast will remain unchanged, i.e. if it is rainy today, the persistence model will predict that it will be rainy tomorrow as well. This method works well when weather patterns change slowly or weather is in a steady state, such as during the summer season in the tropics. This not only applies to short-term forecasting, but also for long range weather conditions (e.g. monthly predictions). In the last years, rapid Artic warming and uncommonly stationary waves of the jet stream have been seen to favour these persistent weather patterns [1].

In some cases, persistence models are "hard to beat" by more sophisticated weather forecasting methods because of the stagnant dynamic of the system. This can happen even when information about the previous states is included in the prediction model. As an alternative to this problem, this paper proposes the use of a strategy referred to in the literature as mixture of experts (ME), based on the divide and conquer principle. ME approaches assign different regions of the problem space to different experts, which are then supervised and managed by a gating function. Usually, the experts and the gating function are learnt together in an optimisation framework. The first expert that we consider is the persistence model, which already successfully predicts the next state of the system for most cases. The second expert is a machine learning model that predicts the output of the system when the persistence model is not accurate. Our model does not attempt to simply beat the persistence model, but rather assumes this persistent behaviour as a baseline and complement its performance when more drastic changes occur.

The problem considered in this paper is that of predicting low-visibility events at airports. Air transportation is probably the most affected sector by foggy and misty periods, since these events can dramatically restrict airport use and cause flight delays, diversions and cancellations [2], or accidents in the worst cases [3]. The meteorological services that support air navigation systems prepare and disseminate terminal aerodrome forecasts to support the aeronautical community when dealing with airport low-visibility conditions. These forecasts are used for pre and intra-flight planning and can help air traffic managers to activate procedures to ensure safe air operations during these events. However, forecasting low-visibility conditions is not an easy task, mainly because fog formation is very sensitive to small-scale variations of atmospheric variables. For this reason, aeronautical meteorological forecasters need normally to integrate different sources of information to provide a robust prediction of low-visibility events. Recently, machine learning methods are gaining popularity for this task and they are being used to help forecasters improve the prediction of reduced visibility events at airports facilities [4], [5], [6].

We evaluate the performance of our ME machine-learning model to forecast the runway visual range and cloud height at the airport, both of which are crucial variables to determine low-visibility conditions. Contrary to previous approaches, the problem is tackled as an ordinal classification problem by discretising the time series in different categories. Given that four categories or ranges are enough for obtaining practical information, the main advantage of using this discretisation

is the corresponding simplification of the prediction problem. The order of the categories implies the use of ordinal classifiers [7], which are specifically designed for minimising the deviation of the predicted categories from the actual ones.

The methodology is tested with data collected at the Valladolid airport (Spain), which shows a persistence of approximately 90% for hourly prediction. We evaluate the performance of our models at different time spans and with different window sizes. The prediction is performed using 7 atmospheric variables. Our results show that persistence model can be successfully complemented by machine learning, leading to a superior performance, specifically for long-term prediction, which is indeed necessary for successful airport managing.

The rest of the paper is structured as follows: Section II introduces the proposed mixture of experts, Section III describes the datasets used, the experiments performed and the results obtained, and finally, Section IV outlines some conclusions and future work.

## II. Methodology

This section presents the mixture of experts model proposed.

### A. Previous notions

Our dataset $D$ is composed of a set of weather-related exogenous variables $\mathbf{x}$ and an output label $y$ which contains information about airport visibility, so that $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$. We study the prediction of runway visual range (RVR) and cloud height (CH), both of which are further described in Section III-A. These variables are discretised in classes, given that the application at hand might only requires a coarse-grain prediction and this greatly simplifies the prediction task. A common way of discretising a variable is:

$$y_t = \begin{cases} \mathcal{C}_1, & \text{if } -\infty < r_t < R_1, \\ \mathcal{C}_2, & \text{if } R_1 \leq r_t < R_2, \\ \ldots & \\ \mathcal{C}_Q, & \text{if } R_{Q-1} \leq r_t < -\infty, \end{cases} \quad (1)$$

where $r_t$ is the real value being observed, and $R_1, \ldots, R_{Q-1}$ are a set of thresholds defined by the experts. The classes then are known to follow a specific order of the form $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \ldots \prec \mathcal{C}_q$, which can be tackled by a machine learning paradigm know as ordinal classification. This order derives a corresponding misclassification cost, since one aims to minimise the misclassification between classes further apart in the scale.

### B. Proposed mixture of experts

Our aim is to learn a model that complements the persistence in the cases when weather patterns are not stationary. Although we propose a probabilistic model, with a different probability equation for each class, we will start defining it as a regression model to ease its understanding. The mixture of experts that we look for takes the following the form:

$$y_{t+k} = \alpha(\mathbf{z}_t) \cdot f_1(\mathbf{z}_t) + (1 - \alpha(\mathbf{z}_t)) \cdot f_2(\mathbf{z}_t), \quad k \geq 1 \quad (2)$$

where the function $\alpha$ is a probabilistic gating function that decides whether we want to use $f_1$ or $f_2$ (i.e. the experts) to predict the output at $y_{t+k}$, and $k$ is the prediction horizon. The prediction is made based on $\mathbf{z}_t$, which we define as $\{\mathbf{x}_{t-\Delta}, y_{t-\Delta}, \ldots, \mathbf{x}_t, y_t\}$, where $\Delta$ is the window size, i.e. the number of instants we use for the prediction.

In our case, since we know that the underlying dynamic of our system is mostly stationary, we define $f_1$ as the persistence model, i.e. $\hat{y}_{t+k} = f_1(\mathbf{z}_t) = y_t$. $f_2$ is then defined as an expert on the patterns that $f_1$ fails to predict. Eq. 2 can be thus rewritten as:

$$y_{t+k} = \alpha(\mathbf{z}_t) \cdot y_t + (1 - \alpha(\mathbf{z}_t)) \cdot f_2(\mathbf{z}_t), \quad (3)$$

where $y_t$ is known and $\alpha$ and $f_2$ needs to be optimised. We define $f_2$ as an autoregressive neural network and $\alpha$ as an autoregressive logistic regression function. Both functions can be optimised together through gradient descent.

Given that we are dealing with a classification problem, we need to adapt Eq. 3 and separately estimate the probability that a pattern $y_{t+k}$ belongs to $\mathcal{C}_q$:

$$P(y_{t+k} = \mathcal{C}_q | \mathbf{z}_t, \boldsymbol{\nu}, \boldsymbol{\kappa}) = \alpha(\mathbf{z}_t, \boldsymbol{\nu}) \cdot [[y_t = \mathcal{C}_q]] + \ldots \quad (4)$$
$$+(1 - \alpha(\mathbf{z}_t, \boldsymbol{\nu})) \cdot P_{\text{net}}(y_{t+k} = \mathcal{C}_q | \mathbf{z}, \boldsymbol{\kappa}),$$

where $[[\cdot]]$ is a boolean test returning 1 if the condition is true (0 otherwise) and represents the persistence for each individual probability, $P_{\text{net}}(y_{t+k} = \mathcal{C}_q | \mathbf{z}_t, \boldsymbol{\kappa})$ is a probabilistic autoregressive neural network (which will be further detailed in Section II-D) with parameters $\boldsymbol{\kappa}$, and $\alpha(\mathbf{z}_t, \boldsymbol{\nu})$ is a probabilistic autoregressive logistic regression model with parameters $\boldsymbol{\nu}$, i.e.:

$$\alpha(\mathbf{z}_t, \boldsymbol{\nu}) = \frac{1}{1 + \exp(-\boldsymbol{\nu}^T \cdot (1, \mathbf{z}_t))} = \sigma(\boldsymbol{\nu}^T \cdot (1, \mathbf{z}_t)),$$

where $\sigma(x)$ is the sigmoid function. The class predicted will be given the maximum a posteriori probability:

$$\hat{y}_{t+k} = \arg\max_{\mathcal{C}_q} P(y_{t+k} = \mathcal{C}_q | \mathbf{z}_t, \boldsymbol{\nu}, \boldsymbol{\kappa}).$$

### C. Learning the free parameters

We propose two schemes for optimising the parameters $\boldsymbol{\kappa}$ and $\boldsymbol{\nu}$, which are detailed below.

*1) Independent training:* The most simple strategy for optimising these parameters is to train them independently following these steps:

1) Run $\hat{y}_{t+k} = y_t$ (i.e. compute $f_1$) through $D$.
2) Identify problematic cases $\mathbf{Z} = \{\mathbf{z}_{n_1}, \ldots, \mathbf{z}_{n_\text{p}}\}$, i.e. patterns for which the persistence model does not predict the result accurately such that $y_{t+k} \neq y_t$.
3) Define $C = \{(\mathbf{z}_1, c_1), \ldots, (\mathbf{z}_N, c_N)\}$, where $c_i$ is defined as 0 if the pattern is problematic, and 1 otherwise.
4) Learn $\alpha(\mathbf{z}_t, \boldsymbol{\nu})$ using $C$, which is a binary problem. We consider a standard logistic regression learner.
5) Train $f_2$ only on problematic cases, $P = \{(\mathbf{z}_{n_1}, y_{n_1+k}), \ldots, (\mathbf{z}_{n_1}, y_{n_\text{p}+k})\}$. The error function and the neural network model used are similar to the ones explained in the next subsection.

This strategy focuses the neural network component only on problematic cases and does not consider potential interactions between both the logistic regression model and the network.

*2) Simultaneous training:* To obtain the best potential of this mixture of experts, it is desirable to optimise both models $\alpha$ and $f_2$ together. A convenient strategy is thus to apply gradient descent over the whole parameter vector $\mathbf{s} = \{\boldsymbol{\nu}, \boldsymbol{\kappa}\}$ with the training set $D$.

The cross-entropy function can be minimised for this purpose:

$$L_{\mathrm{O}}(\mathbf{s}, D) = -\frac{1}{N}\sum_{t=1}^{N}\sum_{q=1}^{Q}[[y_{t+k} = \mathcal{C}_q]] \log p_{tq},$$

where $p_{tq} = P(y_{t+k} = \mathcal{C}_q|\mathbf{z}_t, \boldsymbol{\nu}, \boldsymbol{\kappa})$ is the estimation of probability given by Eq. 4. The gradient descent technique used is the $iRprop^+$ algorithm, which usually provides robust performance [8].

Given that the datasets are moderately imbalanced (see Section III-A), we also include different costs for the classes of the problem, according to their a priori probability:

$$L_{\mathrm{W}}(\mathbf{s}, D) = -\frac{1}{N}\sum_{t=1}^{N}\sum_{q=1}^{Q} o_q[[y_{t+k} = \mathcal{C}_q]] \log p_{tq},$$

where $o_q = 1 - \frac{N_q}{N}$, and $N_q$ is the number of patterns of class $\mathcal{C}_q$. We also include $L2$ regularisation in the error function to avoid overfitting, so that the final cost is:

$$L(\mathbf{s}, D) = L_{\mathrm{W}}(\mathbf{s}, D) + \lambda \cdot \sum_{i=1}^{S} s_i^2, \tag{5}$$

where $S$ is the total number of parameters and $\lambda$ is the regularisation parameter.

Now we detail the derivatives of the error function with respect to the network parameters. The gradient vector is given by:

$$\nabla L(\mathbf{s}, D) = \left(\frac{\partial L(\mathbf{s}, D)}{\partial s_1}, \frac{\partial L(\mathbf{s}, D)}{\partial s_2}, \ldots, \frac{\partial L(\mathbf{s}, D)}{\partial s_S}\right).$$

Considering Eq. 5, each of the components can be defined as:

$$\frac{\partial L}{\partial s_i} = -\frac{1}{N}\sum_{t=1}^{N}\sum_{q=1}^{Q}\frac{o_q[[y_{t+k} = \mathcal{C}_q]]}{p_{tq}} \cdot \frac{\partial p_{tq}}{\partial s_i} + 2s_i,$$

where $i = 1, \ldots, S$. For the logistic regression parameters, these derivatives are given by:

$$\frac{\partial p_{tq}}{\partial \nu_i} = \sigma'(h)[[y_t = \mathcal{C}_q]] - \sigma'(h) p_{\mathrm{net}nq},$$

where $h = \boldsymbol{\nu}^T \cdot (1, \mathbf{z})$, $\sigma'(x) = \sigma(x)(1 - \sigma(x))$, and $p_{\mathrm{net}nq} = P_{\mathrm{net}}(y_{t+k} = \mathcal{C}_q|\mathbf{z}_t, \boldsymbol{\kappa})$. The specific model for $p_{\mathrm{net}nq}$ and the corresponding derivatives will be detailed in next subsection.

### *D. Proportional Odds Model Neural Network (NNPOM)*

This section defines the neural network component in our model ($f_2$). As stated before, labels are ordered, which encourages the use of an ordinal classification model [9]. In this paper, we adopt a probabilistic framework and consider the Proportional Odds Model Neural Network (NNPOM) presented in previous research [10], [11]. NNPOM is a threshold model, i.e. it approaches ordinal classification by trying to estimate the latent variable originating the different ordinal categories and learning a set of thresholds discretising this variable. Threshold models have been seen to perform well when categories come from a discretised variable [7].

NNPOM extends the Proportion Odds Model (POM) [12], which, at the same time, is an extension of binary logistic regression. NNPOM predicts cumulative probabilities $P(y_{t+k} \preceq \mathcal{C}_j|\mathbf{z}_t)$, which can be used to obtain direct probability estimations as:

$$\begin{aligned} P(y_{t+k} \preceq \mathcal{C}_q|\mathbf{z}_t) &= \sum_{j=1}^{q} P(y_{t+k} = \mathcal{C}_j|\mathbf{z}_t), \\ P(y_{t+k} = \mathcal{C}_q|\mathbf{z}_t) &= P(y_{t+k} \preceq \mathcal{C}_q|\mathbf{z}_t) - \\ &\quad - P(y_{t+k} \preceq \mathcal{C}_{q-1}|\mathbf{z}_t), \end{aligned} \tag{6}$$

with $q = 2, \ldots, Q$, and considering by definition that $P(y_{t+k} = \mathcal{C}_1|\mathbf{z}_t) = P(y_{t+k} \preceq \mathcal{C}_1|\mathbf{z}_t)$ and $P(y_{t+k} \preceq \mathcal{C}_Q|\mathbf{z}_t) = 1$. NNPOM assumes a logistic function for the distribution of the random error component of the latent variable, giving rise to:

$$P(y_{t+k} \preceq \mathcal{C}_q|\mathbf{z}_t) = \sigma(f(\mathbf{z}_t, \boldsymbol{\theta}) - b_q), \tag{7}$$

where $q = 2, \ldots, Q-1$, $b_q$ is the threshold for class $\mathcal{C}_q$, and $f(\mathbf{z}_t, \boldsymbol{\theta})$ is the projection of the model for pattern $\mathbf{z}_t$. The following constraints must be satisfied: $b_1 \leq \ldots \leq b_{Q-1}$, in order to ensure the monotonicity of the cumulative probabilities. However, we can apply unconstrained optimisers by defining the thresholds as:

$$b_q = b_1 + \sum_{j=2}^{q} a_j^2$$

with padding variables $a_j$, which are squared to make them positive, and $b_1, a_j \in \mathbb{R}$.

Considering Eq. 6 and 7, the probabilities estimated by POM and NNPOM are defined in the following way:

$$\begin{aligned} P(y_{t+k} = \mathcal{C}_1|\mathbf{z}_t) &= \sigma(f(\mathbf{z}_t, \boldsymbol{\theta}) - b_1), \\ P(y_{t+k} = \mathcal{C}_q|\mathbf{z}_t) &= \sigma(f(\mathbf{z}_t, \boldsymbol{\theta}) - b_q) - \sigma(f(\mathbf{z}_t, \boldsymbol{\theta}) - b_{q-1}), \\ &\qquad q \in \{2, \ldots, Q-1\}, \\ P(y_{t+k} = \mathcal{C}_Q|\mathbf{z}_t) &= 1 - \sigma(f(\mathbf{z}_t, \boldsymbol{\theta}) - b_{Q-1}). \end{aligned}$$

The main difference between POM and NNPOM is found in $f(\mathbf{z}_t, \boldsymbol{\theta})$: POM estimates the latent variable as a linear model of the inputs, $f(\mathbf{z}_t, \boldsymbol{\theta}) = \boldsymbol{\theta}^T \cdot (1, \mathbf{z}_t)$, while a linear combination of nonlinear basis functions (hidden neurons) is assumed for NNPOM:

$$f(\mathbf{z}_t, \boldsymbol{\theta}) = \sum_{j=1}^{M} \beta_j B_j(\mathbf{z}_t, \mathbf{w}_j),$$

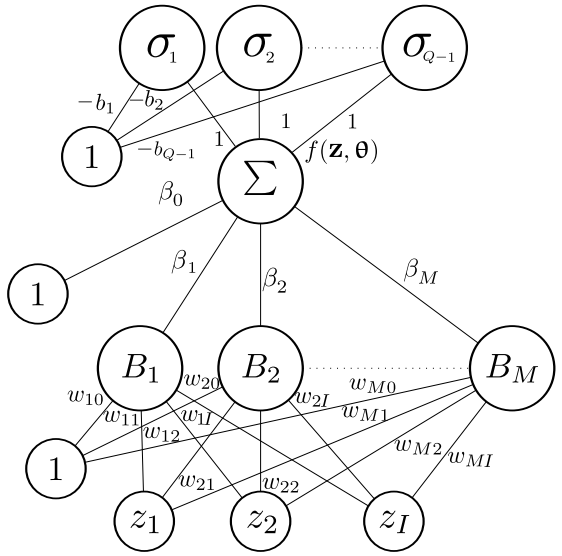


Fig. 1: Structure of the NNPOM model, including one output node with different biases, $M$ hidden nodes and $k$ input nodes

where $M$ is the number of hidden units, $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \mathbf{W}\}$, $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_M\}$, $\mathbf{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_M\}$, and $B_j(\mathbf{z}_t, \mathbf{w}_j)$ can be any type of basis functions, in our case, sigmoidal units, $B_j(\mathbf{z}_t, \mathbf{w}_j) = \sigma(\mathbf{w}_j^T \cdot (1, \mathbf{z}_t))$, where $\mathbf{w}_j = \{w_{j0}, w_{j1}, \ldots, w_{jI}\}$, $I$ is the number of inputs and $w_0$ is the bias. The final parameter vector of NNPOM is defined as $\boldsymbol{\kappa} = \{\boldsymbol{\beta}, \mathbf{W}, b_1, a_2, \ldots, a_{Q-1}\}$. The structure of the NNPOM model is detailed in Fig. 1.

To ease the notation, we define $p_{tq} = P_{\text{net}}(y_{t+k} = \mathcal{C}_q | \mathbf{z}_t, \boldsymbol{\kappa})$, $g_{tq} = g(\mathbf{x}_t, \boldsymbol{\kappa}) = \sigma(f_t - b_q)$ and $f_t = f(\mathbf{x}_t, \boldsymbol{\beta}, \mathbf{W})$. According to Eq. 6, the derivatives of the probability function can be specified by:

$$\begin{aligned}
\frac{\partial p_{t1}}{\partial s_i} &= \sigma'(g_{t1}) \frac{\partial g_{t1}}{\partial s_i}, \\
\frac{\partial p_{tq|1<q<Q}}{\partial s_i} &= \sigma'(g_{tq}) \frac{\partial g_{tq}}{\partial s_i} - \sigma'(g_{t(q-1)}) \frac{\partial g_{t(q-1)}}{\partial s_i}, \\
\frac{\partial p_{tQ}}{\partial s_i} &= -\sigma'(g_{t(Q-1)}) \frac{\partial g_{t(Q-1)}}{\partial s_i},
\end{aligned}$$

where $s_i \in \boldsymbol{\kappa}$.

The derivatives for $g_{tq}$ (i.e. $s_i \in \{\boldsymbol{\beta}, \mathbf{W}\}$) are the standard ones for multilayer perceptrons:

$$\begin{aligned}
\frac{\partial g_{tq}}{\partial \beta_j} = \frac{\partial f_t}{\partial \beta_j} &= B_j(\mathbf{z}_t, \mathbf{w}_j), \\
\frac{\partial g_{tq}}{\partial w_{ji}} = \frac{\partial f_t}{\partial w_{ji}} &= \beta_j \sigma'(\mathbf{w}_j^T \cdot (1, \mathbf{z}_t)) z_{ti},
\end{aligned}$$

where $i \in \{1, \ldots, I\}$, $j \in \{1, \ldots, M\}$, and $I$ is the number of input variables.

For the threshold and padding parameters, the derivatives are:

$$\frac{\partial g_{tq}}{\partial b_1} = -1, \frac{\partial g_{tq}}{\partial a_j} = \begin{cases} 0, & \text{if } q < j, \\ -2a_j, & \text{otherwise.} \end{cases}$$

where $j \in \{2, \ldots, Q-1\}$.

TABLE I: Number of patterns and class distribution of the datasets for different time horizons ($k$) and window sizes ($\Delta$).

| Dataset: CH | | |
|---|---|---|
| | # Patterns | Distribution |
| $\Delta = 1$, $k = 1$ | 5974 | [4178, 1170, 626] |
| $\Delta = 1$, $k = 3$ | 5261 | [3749, 1017, 495] |
| $\Delta = 1$, $k = 6$ | 4200 | [3156, 747, 297] |
| $\Delta = 3$, $k = 1$ | 5258 | [3746, 1017, 495] |
| $\Delta = 3$, $k = 3$ | 4549 | [3350, 848, 351] |
| $\Delta = 3$, $k = 6$ | 3489 | [2751, 520, 218] |
| $\Delta = 5$, $k = 1$ | 4546 | [3347, 848, 351] |
| $\Delta = 5$, $k = 3$ | 3837 | [2957, 628, 252] |
| $\Delta = 5$, $k = 6$ | 2779 | [2254, 351, 174] |
| Dataset: RVR | | |
| | # Patterns | Distribution |
| $\Delta = 1$, $k = 1$ | 8520 | [7199, 903, 312, 106] |
| $\Delta = 1$, $k = 3$ | 8518 | [7197, 903, 312, 106] |
| $\Delta = 1$, $k = 6$ | 8515 | [7194, 903, 312, 106] |
| $\Delta = 3$, $k = 1$ | 8518 | [7197, 903, 312, 106] |
| $\Delta = 3$, $k = 3$ | 8516 | [7195, 903, 312, 106] |
| $\Delta = 3$, $k = 6$ | 8513 | [7192, 903, 312, 106] |
| $\Delta = 5$, $k = 1$ | 8516 | [7195, 903, 312, 106] |
| $\Delta = 5$, $k = 3$ | 8514 | [7193, 903, 312, 106] |
| $\Delta = 5$, $k = 6$ | 8511 | [7190, 903, 312, 106] |

## III. Experiments

This section presents the performance of the previously presented approaches for low-visibility events prediction and analyses the results obtained.

### A. Data description

The datasets used consider the prediction of low-visibility events at Valladolid airport, Spain (41.70 N, 4.88 W). This airport is well-known for its foggy days. Due to its geographical and climatological characteristics, radiation fog is very frequent [13]. A detailed Valladolid airport climatology can be found in [14].

In order to have in-situ information about the most basic parameters involved in radiation fog events at the airport, we used meteorological data obtained from the two runway thresholds. Landing operating minima are usually expressed in terms of a minimum decision height and a minimum runway visual range (RVR) value. RVR is a meteorological parameter measured at the aerodrome, but decision height is not a meteorological variable that can be estimated, as it is a reference for the pilots to decide whether or not continue with the landing. The closer meteorological variable is cloud height (CH), which can be generally found in aerodrome METAR reports. Consequently, we consider the prediction of RVR and CH at the airport. These two variables are critical to determine the acceptable minima for landing operations under different categories of Instrument Landing System (so-called CAT I, CAT II and CAT III). They are also crucial to help airport managers activate low visibility procedures. To obtain the values of these variables, we use direct measurements from three visibilimeters deployed along the airport runway (touchdown zone, the mid-point and stop-end of the runway).

TABLE II: Mean test results obtained by the different methods compared for CH and RVR and different time horizons (k) and window size ($\Delta$).

| Dataset: CH | Method | $Acc$ | $AMAE$ | $MMAE$ | $GM$ | Dataset: RVR | Method | $Acc$ | $AMAE$ | $MMAE$ | $GM$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\Delta = 1$, $k = 1$ | Persist | *87.36* | **0.2087** | *0.2866* | **81.54** | $\Delta = 1$, $k = 1$ | Persist | 89.80 | **0.3776** | **0.5711** | **63.34** |
| | POM | 87.16 | 0.2323 | 0.3307 | 78.79 | | POM | **90.76** | *0.3808* | *0.5746* | *62.73* |
| | NNPOM | 86.52 | 0.2519 | 0.3817 | 76.81 | | NNPOM | 90.55 | 0.4489 | 0.7927 | 52.66 |
| | ITME | 87.32 | *0.2091* | *0.2866* | *81.52* | | ITME | 89.64 | 0.4643 | 0.9227 | 55.35 |
| | STME | **87.41** | 0.2340 | 0.3475 | 79.63 | | STME | *90.66* | 0.4185 | 0.6444 | 58.76 |
| | STMEIC | 85.93 | 0.2130 | **0.2832** | 81.31 | | STMEIC | 86.98 | 0.3939 | 0.6149 | 61.77 |
| $\Delta = 1$, $k = 3$ | Persist | **77.44** | **0.3759** | **0.5141** | **67.22** | $\Delta = 1$, $k = 3$ | Persist | 83.74 | *0.7239* | *1.2927* | *37.68* |
| | POM | 77.02 | 0.5181 | 0.9767 | 35.63 | | POM | **85.88** | 0.9250 | 1.8635 | 7.42 |
| | NNPOM | 74.98 | 0.5719 | 1.0173 | 43.82 | | NNPOM | 85.29 | 0.8594 | 1.5876 | 31.44 |
| | ITME | 77.17 | *0.4168* | *0.6290* | *64.51* | | ITME | 84.29 | 0.9178 | 1.8055 | 21.04 |
| | STME | *77.32* | 0.5055 | 0.8392 | 55.45 | | STME | *85.49* | 0.8583 | 1.6031 | 32.30 |
| | STMEIC | 77.00 | 0.4222 | 0.7143 | 61.43 | | STMEIC | 80.56 | **0.6849** | **1.2764** | **38.19** |
| $\Delta = 1$, $k = 6$ | Persist | 67.95 | **0.5000** | **0.6588** | **55.04** | $\Delta = 1$, $k = 6$ | Persist | 79.94 | 0.9796 | 1.8381 | 23.13 |
| | POM | *75.36* | 0.7517 | 1.4683 | 0.00 | | POM | **84.46** | 1.1937 | 2.4438 | 0.00 |
| | NNPOM | **75.45** | 0.6630 | 1.2531 | 34.94 | | NNPOM | 84.12 | *0.9296* | 1.7808 | 27.93 |
| | ITME | 72.31 | 0.7285 | 1.4201 | 27.70 | | ITME | 84.09 | 1.0603 | 1.8855 | 19.51 |
| | STME | 73.86 | 0.6302 | 1.1380 | *42.75* | | STME | *84.18* | 0.9332 | *1.7448* | *28.02* |
| | STMEIC | 70.48 | *0.5895* | *1.0624* | 41.58 | | STMEIC | 78.55 | **0.8049** | **1.5195** | **32.02** |
| $\Delta = 3$, $k = 1$ | Persist | 87.36 | 0.2087 | 0.2866 | 81.54 | $\Delta = 3$, $k = 1$ | Persist | 89.80 | **0.3776** | **0.5711** | **63.33** |
| | POM | 87.02 | 0.2370 | 0.3395 | 78.26 | | POM | **91.10** | *0.3882* | *0.5870* | *62.66* |
| | NNPOM | 86.64 | 0.2453 | 0.3775 | 76.45 | | NNPOM | 90.46 | 0.4753 | 0.8415 | 51.10 |
| | ITME | **87.53** | *0.1959* | *0.2728* | *82.11* | | ITME | 89.40 | 0.5015 | 1.0448 | 50.03 |
| | STME | *87.48* | 0.2216 | 0.3355 | 79.72 | | STME | *90.84* | 0.4345 | 0.7053 | 57.23 |
| | STMEIC | 86.03 | **0.1939** | **0.2469** | **82.28** | | STMEIC | 86.43 | 0.4159 | 0.6864 | 60.11 |
| $\Delta = 3$, $k = 3$ | Persist | 77.43 | *0.3760* | **0.5141** | **67.22** | $\Delta = 3$, $k = 3$ | Persist | 83.74 | 0.7240 | 1.2927 | 37.68 |
| | POM | 77.41 | 0.5041 | 0.9228 | 45.94 | | POM | 86.25 | 0.8956 | 1.8254 | 7.74 |
| | NNPOM | *79.28* | 0.4653 | 0.8050 | 54.68 | | NNPOM | *86.27* | *0.6771* | *1.2622* | **44.30** |
| | ITME | 76.71 | 0.4810 | 0.8944 | 52.07 | | ITME | 84.75 | 0.8585 | 1.6475 | 24.33 |
| | STME | **79.81** | 0.4339 | 0.7271 | 60.38 | | STME | **86.60** | 0.6890 | 1.3021 | 41.70 |
| | STMEIC | 77.16 | **0.3556** | *0.5476* | *67.21* | | STMEIC | 81.39 | **0.6002** | **1.1624** | *42.08* |
| $\Delta = 3$, $k = 6$ | Persist | 67.95 | *0.5000* | **0.6588** | **55.04** | $\Delta = 3$, $k = 6$ | Persist | 79.94 | 0.9797 | 1.8381 | 23.13 |
| | POM | 75.73 | 0.7147 | 1.3994 | 12.45 | | POM | 84.47 | 1.1285 | 2.3229 | 0.00 |
| | NNPOM | **80.40** | 0.5284 | 0.9666 | 50.90 | | NNPOM | 84.31 | 0.7569 | 1.3973 | 41.15 |
| | ITME | 76.02 | 0.6622 | 1.2550 | 32.95 | | ITME | *84.52* | 0.9720 | 1.7125 | 27.03 |
| | STME | *79.51* | 0.5535 | 0.9893 | 47.64 | | STME | **85.14** | *0.7347* | *1.3259* | **43.41** |
| | STMEIC | 71.93 | **0.4798** | *0.7766* | *52.74* | | STMEIC | 78.29 | **0.6632** | **1.2319** | *43.16* |
| $\Delta = 5$, $k = 1$ | Persist | *87.35* | **0.2088** | **0.2866** | **81.53** | $\Delta = 5$, $k = 1$ | Persist | 89.80 | **0.3776** | **0.5711** | **63.33** |
| | POM | 86.73 | 0.2417 | 0.3448 | 77.79 | | POM | **90.81** | *0.4040* | *0.6278* | *60.64* |
| | NNPOM | 86.77 | 0.2724 | 0.4476 | 72.89 | | NNPOM | 90.16 | 0.4843 | 0.8467 | 51.15 |
| | ITME | **87.92** | *0.2139* | *0.3076* | *80.97* | | ITME | 89.35 | 0.5316 | 1.1571 | 46.47 |
| | STME | 87.35 | 0.2741 | 0.4578 | 75.13 | | STME | *90.74* | 0.4398 | 0.7318 | 56.88 |
| | STMEIC | 85.37 | 0.2268 | 0.3314 | 78.97 | | STMEIC | 86.26 | 0.4309 | 0.7244 | 58.19 |
| $\Delta = 5$, $k = 3$ | Persist | 77.42 | **0.3760** | **0.5141** | **67.21** | $\Delta = 5$, $k = 3$ | Persist | 83.73 | 0.7240 | 1.2927 | 37.68 |
| | POM | 77.71 | 0.5023 | 0.8952 | 48.51 | | POM | 86.28 | 0.8737 | 1.7775 | 8.04 |
| | NNPOM | **81.80** | 0.4801 | 0.8773 | 54.26 | | NNPOM | *87.10* | *0.6217* | *1.1599* | **50.13** |
| | ITME | 78.02 | 0.5554 | 1.1021 | 39.50 | | ITME | 85.10 | 0.8498 | 1.6264 | 26.04 |
| | STME | *81.36* | 0.4795 | 0.8542 | 53.76 | | STME | **87.26** | 0.6247 | **1.1571** | *48.72* |
| | STMEIC | 77.33 | *0.3786* | *0.6406* | *63.38* | | STMEIC | 81.50 | **0.6129** | 1.2036 | 41.11 |
| $\Delta = 5$, $k = 6$ | Persist | 67.95 | 0.5000 | **0.6588** | *55.04* | $\Delta = 5$, $k = 6$ | Persist | 79.93 | 0.9796 | 1.8381 | 23.13 |
| | POM | 75.47 | 0.7307 | 1.4406 | 12.41 | | POM | 84.90 | 1.1021 | 2.2790 | 0.00 |
| | NNPOM | *82.78* | 0.5035 | 0.9189 | 52.64 | | NNPOM | *85.10* | 0.7345 | 1.3738 | *43.22* |
| | ITME | 77.46 | 0.6819 | 1.2438 | 34.67 | | ITME | 84.40 | 0.9835 | 1.7932 | 27.56 |
| | STME | **83.53** | *0.4818* | 0.8678 | 54.80 | | STME | **85.58** | *0.7204* | *1.3403* | **45.40** |
| | STMEIC | 75.38 | **0.4630** | *0.7646* | **55.44** | | STMEIC | 78.15 | **0.6650** | **1.2552** | 42.72 |

These instruments are part of the Meteorological State Agency of the Spanish aeronautical observation network. The complete list of input variables considered in this study are the same for both datasets: hour, temperature in Celsius, relative humidity (%), wind speed (in KT) and direction (in sexagesimal degrees true) in both runway thresholds (23 and 5 respectively), and atmospheric pressure in hPa. We consider data at the Valladolid airport from winter months (November, December, January and February) of three periods (2009-2010, 2010-2011 and 2011-2012). Hourly values of all the variables are subsequently analysed in this study.

The discretisation of both variables (RVR and CH) follows

the simple scheme of Eq. 1, where the thresholds used for RVR are $R_1 = 300$m, $R_2 = 550$m and $R_3 = 2000$m, resulting in four categories. On the other hand, for CH, the discretisation thresholds are $R_1 = 200$m and $R_2 = 1500$m, which results in three categories. Note that visibilimeters only deliver precise RVR values when this parameter falls under 2000m. Otherwise the system codifies RVR values as 2000m. This further motivated the use of ordinal classifiers, as the corresponding regression problem would be ill-posed.

According with this discretisation, the number of patterns for the two datasets is included in Table I along with the class distribution. RVR data are measured with visibilimeters located along the runway, while, for the estimation of CH, human intervention is needed. This means that RVR is fully available every hour, but the CH information is only available when the airport is open (Valladolid airport is not a 24h airport). That is the reason why the number of available data is different for every variable.

### *B. Methods tested*

The experimental validation of the methodologies presented in this paper includes the following methods:

- Persistence model (Persist), i.e. predicting the label observed in $t$ for time $t + k$.
- A probabilistic autoregressive ordinal model (POM) considering different time windows, which include the previously discussed variables (see Section III-A) from time $t - \Delta$ to time $t$.
- The NNPOM method described in Section II-D with the same autoregressive structure. All the methods proposed with Mixture of Experts make use of this classification algorithm for the neural network component.
- Independently Trained Mixture of Experts (ITME), as detailed in Section II-C1.
- Simultaneously Trained Mixture of Experts (STME), as described in Section II-C2, without including specific costs for giving more weight to less frequent classes (i.e. $o_q = 1, \forall q$).
- The same STME model but including imbalanced costs (STMEIC).

### *C. Experimental setup*

The time series evaluated includes data from 3 consecutive winters. In order to better validate the methodologies and avoid the dependence of the results on the specific training/test split, we have performed 3 different splits. For each split, the data from one winter forms the test set, while the other two winters are used for training. Average and standard deviation results are provided.

Different problems were derived according to the prediction time horizon (parameter $k$ in Eq. 2, where we set $k = 1$, $k = 3$ and $k = 6$). Moreover, different input windows were compared, depending on the number of steps before included in the independent variables ($\Delta = 1$, $\Delta = 3$ and $\Delta = 5$). Consequently, a total of 18 different datasets were included in our experiments (9 datasets for each variable). Note that depending on the specific setup some data is not available (for example, given that the CH data is available from 5am, a $\Delta = 5$ window means that the first prediction can be done at 10am).

All the models trained by gradient descent methods are stochastic, because the results depend on the initialisation of the parameter vector. Because of this, NNPOM, ITME, STME and STMEIC were run 10 times, and the results reported are the average performance values of the 10 final models.

The architecture and training parameters of the neural network models (number of hidden nodes $M$, regularisation parameter $\lambda$ and maximum number of iterations $iter$) have a decisive impact on the performance of the model. Optimal values can vary for each dataset and even for different training/test splits. The most reliable way of fitting these parameters without favouring any method is applying a nested cross-validation procedure and repeating the training process using the value resulting in the best validation performance. In this way, for NNPOM, ITME, STME and STMEIC, a 5-fold cross-validation model selection was applied, where the ranges explored were: $M \in \{5, 10, 25, 50, 75\}$, $iter \in \{100, 250, 500, 1000\}$, $\lambda \in \{0, 0.001\}$ (preliminary experiments concluded that, for all datasets, higher regularisation rates always led to worse results).

### *D. Performance evaluation*

The following performance metrics have been considered in the comparison of models:

- The accuracy ($Acc$) is defined by:

$$Acc = \frac{100}{N} \sum_{i=1}^{N} [[\hat{y}_i = y_i]],$$

  where $y_i$ is the desired output for time instant $i$, $\hat{y}_i$ is the prediction of the model and $N$ is the total number of patterns in the dataset.
- The Mean Absolute Error ($MAE$) is a common metric for ordinal classification problems which represents the average deviation in absolute value of the predicted class from the true class (considering the order of the classes in the scale). According to [15], this measure should modified in imbalanced datasets, by taking the relative frequency of the classes into account. In this way, we have evaluated the Average $MAE$ ($AMAE$) and Maximum $MAE$ ($MMAE$) :

$$AMAE = \frac{1}{Q} \sum_{q=1}^{Q} MAE_q = \frac{1}{Q} \sum_{q=1}^{Q} \frac{1}{N_q} \sum_{i=1}^{N_q} e_i,$$

$$MMAE = \frac{1}{Q} \max_{q=1}^{Q} MAE_q,$$

  where $e_i = |\mathcal{O}(y_i) - \mathcal{O}(\hat{y}_i)|$ is the distance between the true and the predicted ranks, and $\mathcal{O}(\mathcal{C}_q) = q$ is the position of the $q$-th label. $AMAE$ values range from 0 to $Q - 1$, and so do $MMAE$ values.

- Finally, the geometric mean of the sensitivities of each class ($GMS$) is a summary of the percentages of correct classification individually obtained for each class:

$$GMS = \sqrt[Q]{\prod_{q=1}^{Q} S_q},$$

where the sensitivities, $S_q$, are obtained as:

$$S_q = \frac{100}{N_q} \sum_{i=1}^{N_q} [[\hat{y}_i = y_i]], q \in \{1, \ldots, Q\},$$

where $N_q$ represents the number of patterns of class $\mathcal{C}_q$. This metric is also a standard for imbalanced problems.

TABLE III: Ranking results according to the predictive variable considered (both, CH or RVR). The results for all prediction horizons are averaged.

| CH and RVR | | | | |
|---|---|---|---|---|
| Method | $Acc$ | $AMAE$ | $MMAE$ | $GM$ |
| Persist | 4.22 | *2.28* | *2.14* | **2.17** |
| POM | *2.89* | 4.94 | 4.94 | 5.06 |
| NNPOM | *2.89* | 4.06 | 4.06 | 3.94 |
| ITME | 3.56 | 4.50 | 4.47 | 4.39 |
| STME | **1.72** | 3.44 | 3.44 | 3.17 |
| STMEIC | 5.72 | **1.78** | **1.94** | *2.28* |
| CH | | | | |
| Method | $Acc$ | $AMAE$ | $MMAE$ | $GM$ |
| Persist | 3.78 | **1.67** | **1.39** | **1.33** |
| POM | 3.89 | 5.22 | 5.22 | 5.44 |
| NNPOM | *3.00* | 4.67 | 4.67 | 4.56 |
| ITME | *3.00* | 3.78 | 3.83 | 3.78 |
| STME | **1.89** | 3.78 | 3.89 | 3.67 |
| STMEIC | 5.44 | *1.89* | *2.00* | *2.22* |
| RVR | | | | |
| Method | $Acc$ | $AMAE$ | $MMAE$ | $GM$ |
| Persist | 4.67 | *2.89* | *2.89* | 3.00 |
| POM | *1.89* | 4.67 | 4.67 | 4.67 |
| NNPOM | 2.78 | 3.44 | 3.44 | 3.33 |
| ITME | 4.11 | 5.22 | 5.11 | 5.00 |
| STME | **1.56** | 3.11 | 3.00 | *2.67* |
| STMEIC | 6.00 | **1.67** | **1.89** | **2.33** |

### *E. Results and discussion*

The mean test results obtained by all the methods compared in this paper can be found in Table II. Moreover, in order to better summarise these results, Tables III and IV show the test mean rankings in terms of all metrics for all the methods considered in the experiments. For each dataset, a ranking of 1 is given to the best method and a 6 is given to the worst one. More specifically, Table III shows the ranking for the different problems, CH and RVR, and Table IV shows the results for the considered prediction horizons.

Several conclusions can be drawn from these results:

- Table II shows how stagnant these variables are for the two considered datasets (e.g. RVR being steady 90% of the time for k=1). Moreover, it can be seen how the prediction of these variables using the persistence model deteriorates for larger time horizons.
- Both Tables III and IV show the difficulty in getting a trade-off between all considered metrics, specifically $Acc$ and the rest of metrics.
- The obtained results in Table II can be said to be generally satisfactory. For both datasets, we obtain relatively low error in ordinal metrics, such as $AMAE$ and $MMAE$, and good performance both in $Acc$ and $GM$. The performance is usually better for CH than for RVR, which may be because the prediction problem is simpler as there is one class less and the dataset is less imbalanced. Even although both datasets are imbalanced, in most cases no class is completely misclassified ($GM = 0.00$).
- Note that the performance gain for larger window sizes is not very high (comparing different $\Delta$ values). This is crucial for CH, given that, the unavailability of results until airport is open limits the first hour when predictions can be obtained, specially for large window sizes.
- Comparing POM and NNPOM to Persist in Table III, it can be seen that in some cases ML models struggle to reach the performance of the persistence. However, NNPOM generally outperforms POM, as expected.
- The mixture of models using an independent optimisation of the free parameters (ITME) does not achieve satisfactory results, since in most cases it deteriorates the performance of the persistence model. This can be due to the imbalanced nature of the binary problem solved by the logistic regression (predicting problematic cases), which biases the final model towards the persistence.
- The mixture of models that use a simultaneous training shows, however, outstanding performance, being competitive against the persistence model, specially for larger time horizons. The model that includes the imbalanced costs generally presents the best results for $AMAE$, $MMAE$ and $GM$, while the model without costs is competitive in $Acc$. In general, the differences favouring them are higher for the RVR variable, as it is a more difficult problem and there is more room for improvement (see Table III). Larger prediction horizons ($k = 3$ and $k = 6$) are also the best scenarios for our proposals (see Table IV), as the persistence obtains worse results in these cases.

The RVR labels obtained by the STMEIC are presented in Fig. 2, compared with target ones and those obtained by POM. As can be seen, the predictions follow the general tendency of the real values, resulting in an acceptably accurate notion of the visibility. STMEIC presents a better prediction than POM for extreme values ($\mathcal{C}_4$) and small fluctuations of visibility.

## IV. Conclusions

This paper presents a mixture of experts model for predicting ordinal categories associated to low-visibility atmospheric events. Given that these patterns are often persistent in time, the model combines an expert predicting the previous category with an autoregressive neural network expert correcting the

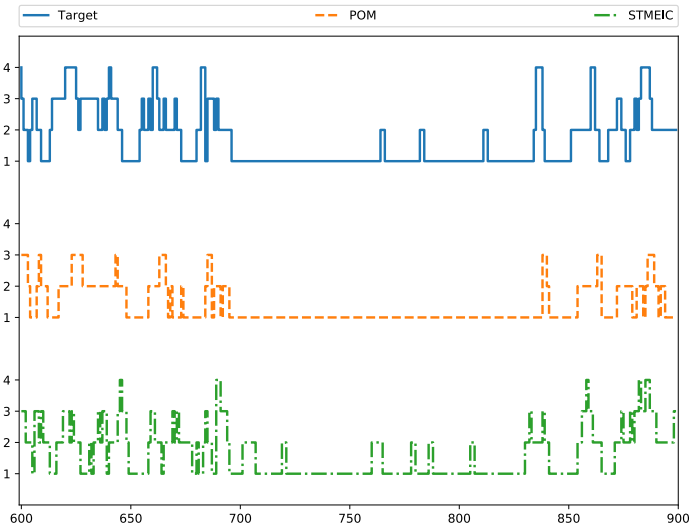


Fig. 2: Test target labels for a range of the RVR time series (time horizon $k = 3$) and labels predicted by STMEIC and POM (window width $\Delta = 3$).

TABLE IV: Ranking results according to the prediction horizon ($k$). Both variables (CH and RVR) are averaged.

| $k = 1$ | | | | |
|---|---|---|---|---|
| Method | *Acc* | *AMAE* | *MMAE* | *GM* |
| Persist | 3.17 | **1.33** | **1.58** | **1.33** |
| POM | *2.67* | 3.17 | 3.17 | 3.33 |
| NNPOM | 3.83 | 5.33 | 5.33 | 5.67 |
| ITME | 3.33 | 4.00 | 4.08 | 3.83 |
| STME | **2.00** | 4.50 | 4.50 | 4.17 |
| STMEIC | 6.00 | *2.67* | *2.33* | *2.67* |
| $k = 3$ | | | | |
| Method | *Acc* | *AMAE* | *MMAE* | *GM* |
| Persist | 4.00 | *2.33* | **2.00** | **2.17** |
| POM | 3.17 | 5.67 | 5.67 | 5.83 |
| NNPOM | *2.67* | 3.67 | 3.50 | 3.00 |
| ITME | 4.00 | 4.67 | 4.67 | 4.67 |
| STME | **1.50** | 3.17 | 3.17 | 3.17 |
| STMEIC | 5.67 | **1.50** | **2.00** | **2.17** |
| $k = 6$ | | | | |
| Method | *Acc* | *AMAE* | *MMAE* | *GM* |
| Persist | 5.50 | 3.17 | 2.83 | 3.00 |
| POM | 2.83 | 6.00 | 6.00 | 6.00 |
| NNPOM | *2.17* | 3.17 | 3.33 | 3.17 |
| ITME | 3.33 | 4.83 | 4.67 | 4.67 |
| STME | **1.67** | *2.67* | *2.67* | *2.17* |
| STMEIC | 5.50 | **1.17** | **1.50** | **2.00** |

persistence when changes are detected. The model is designed from a probabilistic perspective, where the neural network component is based on the proportional odds structure. A gating function, implemented through an autoregressive logistic regression model, assigns the importance of each component.

The model is tested for the task of predicting low-visibility in airports, where the visibility level is represented by two different ordinal categorical variables (cloud height and runway visual height). A battery of experiments is considered, where each variable is evaluated with three different time horizons and three different window widths. Our results are promising, showing very good performance for larger time horizons.

As future research lines, we plan to use more complex neural network models with a recurrent structure to better uncover the dynamics of the time series. Moreover, the training algorithm could be redesigned to alternatively optimise the logistic regression component and the neural network component, in order to accelerate the convergence.

## ACKNOWLEDGEMENT

This work has been subsidized by the projects TIN2014-54583-C2-1-R, TIN2014-54583-C2-2-R, TIN2017-85887-C2-1-P, TIN2017-85887-C2-2-P and TIN2015-70308-REDT of the Spanish Ministry of Economy and Competitiveness (MINECO), FEDER funds (FEDER EU) and the P11-TIC-7508 project of the Junta de Andalucía (Spain).

## REFERENCES

[1] J. A. Francis and S. J. Vavrus, "Evidence for a wavier jet stream in response to rapid arctic warming," *Environmental Research Letters*, vol. 10, no. 1, p. 014005, 2015. [Online]. Available: http://stacks.iop.org/1748-9326/10/i=1/a=014005

[2] J. J. Rebollo and H. Balakrishnan, "Characterization and prediction of air traffic delays," *Trans Res C: Emerging Technol*, vol. 44, pp. 231–241, 2014.

[3] M. M. Ahmed, M. Abdel-Aty, J. Lee, and R. Yu, "Real-time assessment of fog-related crashes using airport weather data: A feasibility analysis," *Accident Anal Prev*, vol. 72, pp. 309–317, 2014.

[4] R. O. Colabone, A. Ferrari, F. da Silva-Vecchia, and A. Bruno-Tech, "Application of artificial neural networks for fog forecast," *J Aerospace Technol Manag*, vol. 7, no. 2, pp. 240–246, 2015.

[5] D. Dutta and S. Chaudhuri, "Nowcasting visibility during wintertime fog over the airport of a metropolis of India: decision tree algorithm and artificial neural network approach," *Nat Hazards*, vol. 75, pp. 1349–1368, 2015.

[6] M. Pérez-Ortiz, S. Jiménez-Fernández, P. A. Gutiérrez, E. Alexandre, C. Hervás-Martínez, and S. Salcedo-Sanz, "A review of classification problems and algorithms in renewable energy applications," *Energies*, vol. 9, no. 8, 2016.

[7] P. A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, F. Fernandez-Navarro, and C. Hervás-Martínez, "Ordinal regression methods: survey and experimental study," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 1, pp. 127–146, 2016.

[8] C. Igel and M. Hüsken, "Empirical evaluation of the improved Rprop learning algorithms," *Neurocomputing*, vol. 50, no. 6, pp. 105–123, 2003.

[9] P. A. Gutiérrez and S. García, "Current prospects on ordinal and monotonic classification," *Progress in Artificial Intelligence*, vol. 5, no. 3, pp. 171–179, 2016.

[10] P. A. Gutiérrez, P. Tiňo, and C. Hervás-Martínez, "Ordinal regression neural networks based on concentric hyperspheres," *Neural Networks*, vol. 59, pp. 51–60, 2014.

[11] M. J. Mathieson, "Ordinal models for neural networks," in *Proceedings of the Third International Conference on Neural Networks in the Capital Markets*, ser. Neural Networks in Financial Engineering, J. M. A.-P. N. Refenes, Y. Abu-Mostafa and A. Weigend, Eds. World Scientific, 1996, pp. 523–536.

[12] P. McCullagh, "Regression models for ordinal data," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 42, no. 2, pp. 109–142, 1980.

[13] C. Román-Gascón, G. Steeneveld, C. Yagüe, M. Sastre, J. Arrillaga, and G. Maqueda, "Forecasting radiation fog at climatologically contrasting sites: evaluation of statistical methods and WRF," *Q J R Meteorol Soc*, vol. 142, pp. 1048–1063, 2016.

[14] AEMET, "Aeronautical climatology of valladolid/villanubla," Agencia Estatal de Meteorologia, Tech. Rep., 2015.

[15] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, and P. A. Gutiérrez, "Metrics to guide a multi-objective evolutionary algorithm for ordinal classification," *Neurocomputing*, vol. 135, pp. 21–31, July 2014.