A Regime-Switching Recurrent Neural Network Model Applied to Wind Time Series

Nikolay Y. Nikolaev
Department of Computing
Goldsmiths College, University of London
London SE14 6NW
United Kingdom
N.Nikolaev@gold.ac.uk

Evgueni Smirnov
Department of Knowledge Engineering
Maastricht University
Maastricht 6200
The Netherlands
Smirnov@maastrichtuniversity.nl

Daniel Stamate
Department of Computing
Goldsmiths College, University of London
London SE14 6NW
United Kingdom
D.Stamate@gold.ac.uk

Robert Zimmer
Department of Computing
Goldsmiths College, University of London
London SE14 6NW
United Kingdom
R.Zimmer@gold.ac.uk

Abstract

This paper proposes a regime-switching recurrent network model (RS-RNN) for non-stationary time series. The RS-RNN model emits a mixture density with dynamic nonlinear regimes that fit flexibly data distributions with non-Gaussian shapes. The key novelties are: development of an original representation of the means of the component distributions by dynamic nonlinear recurrent networks, and derivation of a corresponding expectation maximization (EM) training algorithm for finding the model parameters. The elaborated switching dynamic nonlinear regimes make the RS-RNN especially attractive for describing non-stationary environmental time series. The results show that the RS-RNN applied to a real-world wind speed time series achieves standardized residuals similar to popular previous models, but it is more accurate distribution forecasting than other linear switching (MS-AR) and nonlinear neural network (MLP and RNN) models.

1 Introduction

Regime-switching (RS) models [18] become increasingly popular in practice for density forecasting in environmental sciences. They are found useful for modeling the distributions of sequentially arriving data, like meteorological time series [4,6,12,16,20,32], hydrological time series [42], wind power series [1,13,34,38,48], etc.. These switching models are essentially mixture density models that can approximate well data having distributions of various shapes, which makes them especially attractive for fitting real-world time series contaminated by large amount of noise with unknown character [24,40]. Similar tools generating probabilistic predictions along with probabilistic of their uncertainty are the mixture density neural networks [30], and the kernel density models [47].

The application of the switching models to wind speed time series is currently of particular importance for the supply of renewable wind energy and its integration into the power systems. Wind speed forecasts are required to provide information about future wind energy generation and to reduce instabilities in the energy distribution. Finding accurate solutions to the task of short-term wind speed prediction addressed here is also of special interest for improving power plant scheduling and grid operations management [25]. The main difficulty in this task is the continuous fluctuation of the wind speed due to the stochastic character of the atmospheric processes [8,31]. It is known that pressure, frontline passages and cyclonic conditions lead to variabilities, like nonstationarity and sudden changes in atmospheric series [1,34,35]. The problem is to find good descriptions of such time-varying data distribution with which the wind dynamics can be explained and reconstructed adequately. Although a lot of research have been conducted on describing wind series using various statistical, neural network and hybrid approaches [15,17,47], many of them do not have enough potential for capturing complex variabilities beyond simple periodicities.

The statistical approaches to wind speed prediction build models using time-lagged explanatory variables [14,21]. Popular approaches are the linear autoregressive moving average (ARMA) models [8,24,27,48]. Their enhancements include heteroscedastic variance (ARMA-GARCH) models [40,49], and various switching models, like Smooth Transition AutoRegressive (STAR) [35], Self-Exciting Threshold AutoRegressive (SETAR) [35], and Markov Switching AutoRegressive (MS-AR) models [1,13,34,38]. The ARMA-GARCH models are useful for learning series with non-stationary mean as they use effects from the series variance that help to describe volatile dynamics. The STAR are mixture models that simulate the temporal evolution of the data through weighted averaging of the component regimes, so that their proportions depend on the observed past data. The SETAR and MS-AR are even more attractive as they construct the model from alternating regime submodels, thus leading to more versatile representations. While the regime changes in STAR and SETAR are controlled by the lagged values of the series, the shifts in the MS-AR technology is driven by computable external factors which actually may reflect relevant meteorological features. Hybrid compositions of multiple models have been found to outperform some of the individual models, but they are computationally intensive and have characteristics that are difficult to analyze rigorously [17].

The Markov switching models emit mixtures of Gaussians governed by hidden unobservable regime variables, which helps to fit flexibly data distributions with non-Gaussian shapes. These models explain well shifts in time series because their regimes are controlled in a more sophisticated manner through an exogenous switching variable independent from the lagged observations. The regime-switching models evolve as transitions between regimes and their effect on the full mixture is controlled by specific transition probabilities. The environmental research investigates autoregressive MS-AR models that incorporate mainly linear submodels [1,13,34,38]. They are found particularly suitable for tracking environmental processes with temporally varying structure and non-stationary characteristics [2].

The neural network approaches build nonlinear models that feature elaborate relationships between the selected variables. Some neural networks developed for processing wind series include: feedforward multilayer perceptrons (MLP) [5,10,15,25,43] and recurrent neural networks (RNN) [3,7,26,33,44]. These MLP networks take lagged inputs passed with tapped-delay lines. However, their training does not handle directly the time dimension of the data, rather such time-delay neural networks are still trained with static learning algorithms which yields suboptimal results. The RNNs are more convenient for describing time series as they are dynamic models driven not only by external inputs but also by internal context signals that carry temporal information. The rationale is that the context turns the network into a dynamic function that reflects the time-dependencies in sequentially arriving data.

This paper proposes a Regime-Switching Recurrent Neural Network (RS-RNN) for accurate learning of dynamic mixture distributions. The objective is to build a switching model with improved forecasting potential using separate nonlinear neural networks for each distinct regime, such that they represent the specific means of the distributions [37,39]. Since the widely used standard feed-forward neural networks are static models that do not take into account directly the temporal relationships among the data, we consider the dynamic recurrent neural networks (RNN) [46] as mechanisms for inference of time-dependent wind models as suggested by relevant research [7]. During training the RNN become fitted to their full potential of being dynamic machines which learn time-varying functions, and this is their advantage for modeling environmental time series which exhibit fluctuations of changing magnitude.

The RS-RNN provides a novel mixture density model with dynamic nonlinear regimes that can capture adequately time series and generate probabilistic forecasts. There are two main contributions in the presented research: 1) design of an original switching model in which each regime component is made nonlinear and truly dynamic with a recurrent neural network representation of the mean; and 2) development of a maximum likelihood algorithm for estimation of all RS-RNN parameters formulated according to the expectation maximization (EM) algorithm for mixture models [18,29]. This includes formulation of temporal RTRL derivatives [46] for the recurrent network, likelihood gradients for the computation of the transition probabilities and the regime variances. An essential novelty is that the temporal dimension of the model is treated directly during training, while previous regime-switching models that account for spatiotemporal information still treat the parameters with static estimation algorithms.

This study relates the proposed RS-RNN to benchmark linear autoregressive (AR) models, other switching models with linear regimes (STAR, MS-AR) and nonlinear (MLP,RNN) tools on short-term forecasting (from 1 hour up to 3 days ahead) of real-world wind speed time series. The research analyses the statistical characteristics of the standardized residuals to demonstrate the importance of the nonlinearities and the explicit treatment of the temporal dimension of the model. Using bootstrapped one-step ahead forecasts, obtained with a methodology for neural networks [41], it is shown that RS-RNN achieves better density predictions, which suggests that they are especially suitable for non-stationary time series according to the fundamental literature [9,16,18].

This article is organized as follows. Section two introduces nonlinear regime-switching models and the chosen recurrent Elman-type RNN architecture. Section three gives the EM algorithm for RS-RNN estimation. The forecasting approach is explained in section four. Section five presents the experimental results on time series modeling. Finally, a discussion is made and a conclusion is provided.

2 Nonlinear Switching Model Formulation

A regime-switching model [18] includes a number of component submodels with different parameters which reflect alterations in the data characteristics. The shifts between the regimes are controlled by a discrete state variable that follows an unobservable Markov process. Typically the state is assumed a realization of a two-state Markov chain, that is the current state change depends only on the most recent state which carries all the information from the past. The novelty here is elaboration of nonlinear submodels using recurrent neural networks with internal memory processed by dynamical difference equations. Each component represents a time-dependent density function, and their combination yields a mixture model with dynamic nonlinear regimes that can approximate well non-Gaussian time series.

2.1 The Switching Regime Model

This research considers a regression model of time series data $Y_T = \{y_t\}_{t=1}^{T}$ with regime-dependent mean μ_{S_t} and noise variance $\sigma^2_{S_t}$ where S_t denotes the regime (state). The state $S_t \in \{1, 2\}$ is a discrete switching variable representing the active regime at time t. The assumption is that there is a distinct submodel associated with each regime, that is we have $S_t = i$ for $i = 1, 2$ when two regimes are considered. The novel mixture model developed here represents each of the regime mean components with a nonlinear function $f_i(x_t)$ generated by a dynamic recurrent network with autoregressive inputs $x_t = [y_{t-1}, y_{t-2}, ..., y_{t-l}]$ where l is the embedding dimension.

The original nonlinear regime-switching model is given by the following equations:

$$\begin{aligned} y_t &= \mu_{S_t} + \varepsilon_t, \varepsilon_t \sim \mathcal{N}(0, \sigma^2_{S_t}) \\ x_t &= [y_{t-1}, y_{t-2}, ..., y_{t-l}] \\ \mu_{S_t} &= \sum_{i=1}^{2} f_i(x_t) Ind(S_t = i) \\ \sigma^2_{S_t} &= \sum_{i=1}^{2} \sigma^2_i(x_t) Ind(S_t = i) \end{aligned} \tag{1}$$

where μ_{S_t} is the mixture mean, ε_t is a zero-mean noise with mixture variance $\sigma^2_{S_t}$, $f_i(x_t)$ and $\sigma^2_i(x_t)$ are the regime specific means and variances. The subscripts S_t specify dependence on a particular regime, which is pointed to by the indicator function $Ind(S_t = i)$, it returns one when S_t equals i and zero otherwise. The notation $f_{i,t} = f_i(x_t)$ will be used as abbreviation for the mapping $f_i(x_t)$.

The mean μ_{S_t} and variance $\sigma^2_{S_t}$ of the model depend on the unobserved state S_t. The state S_t is conditioned on the past data with a certain regime probability $\rho_i = P(S_t = i|Y_{t-1})$, $0 < \rho_i < 1$. The regime probability ρ_i accounts for the periods of time during which the chain has been in state i. The substitution $Y_{t-1} = \{y_1, y_2, ..., y_{t-1}\}$ denotes the past information arrived up to time t.

The overall model emits a mixture of distributions each weighted by its corresponding regime probability:

$$p(y_t|Y_{t-1}) = \sum_{i=1}^{2} \rho_i g(y_t|S_t = i, Y_{t-1}) \tag{2}$$

where $g(y_t|S_t = i)$ is the density of the t-th observation for the i-th regime.

The probability density function $g(\cdot)$ conditional on the regime is chosen to be Gaussian:

$$g(y_t|S_t, Y_{t-1}) = \frac{1}{\sqrt{2\pi\sigma^2_{S_t}}} \exp\left(-\frac{(y_t - \mu_{S_t})^2}{2\sigma^2_{S_t}}\right) \tag{3}$$

where dependencies on the past inputs are omitted for clarity from the notation.

The latent state variable S_t is assumed to follow a first-order Markov chain specified by transition probabilities. The transition probabilities specify how likely is the shift from one state to another, and thus they navigate the evolution of the regimes. In an 2-state Markov chain the transition probability r_{ij}, $i = 1, 2$, $j = 1, 2$, that the state S_t will take value i depends only on the most recent past state S_{t-1}:

$$r_{ij} = P(S_t = i|S_{t-1} = j, Y_{t-1}) = P(S_t = i|S_{t-1} = j) \tag{4}$$

where the probability r_{ij} should be restricted $0 < r_{ij} < 1$ in order to avoid having an absorbing regime, and all these probabilities should satisfy $\sum_{i=1}^{2} r_{ij} = 1$ [18].

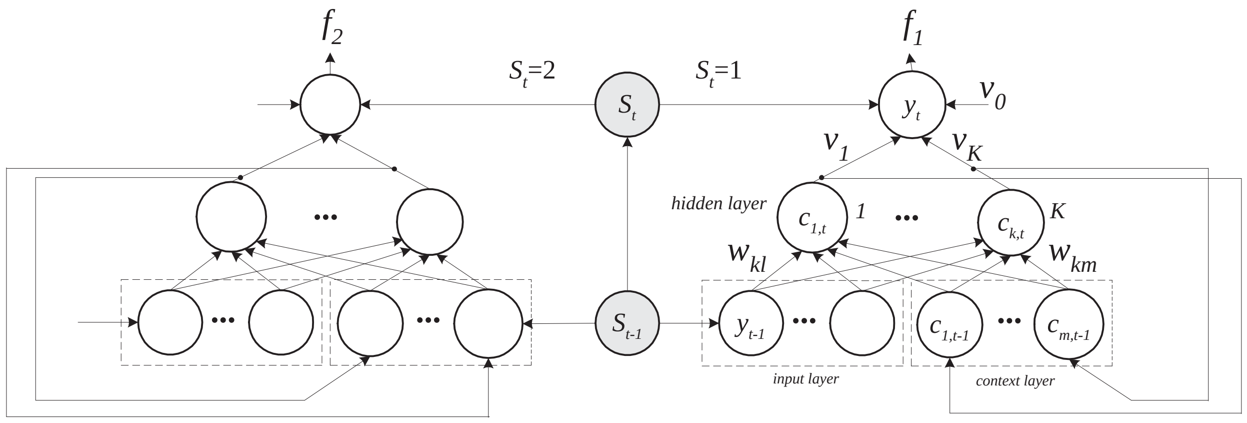

Figure 1. A graphical representation of the relationships between the variables in the regime switching RS-RNN model with Elman topology. The grey nodes are discrete variables, and the remaining are continuous.

2.2 The Elman Recurrent Network

The original idea of this research is to represent the mean of each regime by a separate recurrent network. Here Elman-type RNN [11] is considered as superior to feed-forward networks on temporal processing tasks [23]. The Elman recurrent network has feed-back connections from every hidden node output to every hidden node input via a context layer (Figure 1).

The RNN network is given as input a lagged vector of recent values $x_t = [y_{t-1}, y_{t-2}, ..., y_{t-l}]$. These inputs are processed at the hidden nodes using the hyperbolic tangent activation function φ. The node activations are memorized and serve as context from past information $[c_{1,t}, c_{2,t}, \ldots, c_{K,t}]$, where K is the number of hidden nodes. The output $f_{S_t,t}$ is computed with dynamical equations using not only the given time series data, but also previous information stored in the context layer.

The k-th hidden neuron, $1 \leq k \leq K$, computes one element of the context according to the equation:

$$c_{k,t} = \varphi\left(\sum_{l=1}^{L} w_{kl} y_{t-l} + \sum_{m=1}^{K} w_{km} c_{m,t-1}\right) \tag{5}$$

where w_{kl} are the weights from the l-th input to the k-th hidden node, w_{km} are the weights from the m-th context node (i.e., the feedback signal $c_{m,t-1}$ from the previous step $t-1$) to the k-th hidden node.

The RNN output is produced using the selected activation function φ as follows:

$$f_{S_t,t} = \varphi_{f_{S_t}}\left(\sum_{k=1}^{K} v_k c_{k,t} + v_0\right) \tag{6}$$

where v_k are the weights from the k-th hidden node to the output node, and v_0 is a bias term that feeds 1.

3 RS-RNN Model Estimation

The learning problem [18,19,29] is formulated here as follows: given a training series of data $Y_T = \{y_t\}_{t=1}^{T}$, find the RS-RNN model parameters: transition probabilities, regime network weights and variances $\theta = \left\{\left((r_{i,j})_{j=1}^{2}, \sigma_i^2, [(w_{i,d})_{d=1}^{L+K}, (v_{i,k})_{k=0}^{K}]\right)_{i=1}^{2}\right\}$ which maximize the likelihood that the data have been produced by a generator modeled by the network. Looking for solutions to this problem requires joint search for the regime probabilities of being in a certain regime at a particular time, and for the transition probabilities, weights and variances conditional on the regime characteristics.

The Expectation-Maximization (EM) algorithm [29] is recommended for estimation of mixture models like the RS-RNN as it integrates out efficiently the latent state variable. This algorithm infers the unobserved regimes during the expectation step (treating the parameters as known), and next updates the parameters (transition probabilities, weights and variances) during the maximization step using the inferred regimes. The EM formulation for switching models alternates the two steps seeking iterative maximization of the expectation $E[\log \mathcal{L}]$ of the total negative log-likelihood defined as follows [18]:

$$\log \mathcal{L} = -\sum_{t=1}^{T} \left[\log \sum_{i=1}^{2} \rho_i g(y_t | S_t = i, Y_{t-1}) \right] \tag{7}$$

where $\rho_i = P(S_t = i | Y_{t-1})$ is the regime probability conditioned on the observed data.

3.1 Expectation Step

The E-step [18,29] calculates the expected total log-likelihood $E[\log \mathcal{L}]$ via weighted averaging of the component densities. During this E-step the model parameters $\theta = \left\{\left((r_{i,j})_{j=1}^{2}, \sigma_i^2, [(w_{i,d})_{d=1}^{L+K}, (v_{i,k})_{k=0}^{K}]\right)_{i=1}^{2}\right\}$ are kept fixed, while the regime probabilities are computed using a filtering algorithm [18]. The Hamilton filtering algorithm [18] involves three steps: generating the probabilities of being at each state, combining these probabilities with the information available up to this moment, and updating the probabilities to account for the most recently arrived data.

First, the particular regime probabilities $\rho_i, i = 1, 2$, are predicted using the chain rule for conditional probabilities as follows:

$$\rho_i = P(S_t = i | Y_{t-1}) = \sum_{i=1}^{2} P(S_t = i | S_{t-1} = j) P(S_{t-1} = j | Y_{t-1}) = \sum_{i=1}^{2} r_{ij} P(S_{t-1} = j | Y_{t-1}) \tag{8}$$

where the entries of the transition matrix r_{ij}, $i = 1, 2$, $j = 1, 2$, are set to the following predefined values $P(S_t = 1 | S_{t-1} = 1) = r_{11}$ and $P(S_t = 2 | S_{t-1} = 2) = r_{22}$. Initially we have $P(S_1^1) = (1 - r_{11})/(2 - r_{11} - r_{22})$ and $P(S_1^2) = (1 - r_{22})/(2 - r_{11} - r_{22})$.

Second, the joint density of the observation y_t and the state S_t is obtained as follows:

$$g(y_t, S_t = i | Y_{t-1}) = g(y_t | S_t = i, Y_{t-1}) P(S_t = i | Y_{t-1}) \tag{9}$$

which uses the chosen conditional density $g(y_t | S_t = i, Y_{t-1})$ given that at time t the i-th regime occurs.

Third, the probability of each regime is reevaluated (filtered) to account for the current observation:

$$P(S_t = i|Y_t) = \frac{g(y_t, S_t = i|Y_{t-1})}{\sum_{i=1}^{2} g(y_t, S_t = i|Y_{t-1})} \tag{10}$$

where $g(y_t|Y_{t-1}) = \sum_{i=1}^{2} g(y_t, S_t = i|Y_{t-1})$ is the density of the data.

3.2 Maximization Step

The M-step [18,29] computes the parameters that maximize the expected total log-likelihood $E[\log \mathcal{L}]$ using the inferred regime probabilities. The task to optimize the total log-likelihood suggests to take into account all observations available for learning, before doing parameter estimation. This requires to perform a backward pass over the data in order to improve the regime probabilities using subsequently arrived information, which has not been available during the forward filtering pass.

3.2.1 Backward Smoothing

The smoothed regime probabilities should be conditioned on the full data sample. Efficient smoothing recursions can be obtained according to the Kim's algorithm [22]. It is based on the simplifying assumption that $P(S_t = i|S_{t+1} = j, Y_T) \approx P(S_t = i|S_{t+1} = j, Y_t)$ although using the filtered state transitions may lead to loss of information. Having this approximation allows us to compute efficiently improved smoothed regime probabilities while moving backward over the series with the following equation:

$$P(S_t = i|Y_T) = \sum_{i=1}^{2} \frac{P(S_{t+1} = j|Y_T)P(S_t = i|Y_t)P(S_{t+1} = j|S_t = i)}{P(S_{t+1} = j|Y_t)} \tag{11}$$

which runs backwards in time $t = T-1, ..., 1$. Note that here the conditioning on the data in $P(S_{t+1} = j|Y_t)$ is denoted using $Y_t = \{y_1, y_2, ..., y_t\}$.

3.2.2 Parameter Estimation

Transition Matrix. The state transition matrix contains the probabilities $P(S_{t+1} = j|S_t = i)$ that govern the movement from one regime to another. The entries of this matrix $r_{ij} = P(S_{t+1} = j|S_t = i)$ are obtained with a technique using Lagrange multipliers [18]:

$$r_{ij} = \frac{\sum_{t=2}^{T-1} P(S_{t+1} = j, S_t = i|Y_T)}{\sum_{t=3}^{T} P(S_t = j|Y_T)} \tag{12}$$

where $i = 1, 2$ and $j = 1, 2$ are indices running over the states.

Regime Means. The RNN weights are found by optimization using the derivatives of the expected total log-likelihood. The key idea is to consider separately the log-likelihoods of each regime, that is to maximise $\log \mathcal{L}_i$ conditioned on $S_t = i$ ($\log \mathcal{L}_i = -\sum_{t=1}^{T} \log \mathcal{L}_{i,t}$). Here we compute the likelihood derivatives in the Elman RNN model following the real-time learning (RTRL) algorithm [46] which takes advantage of the full error flow through time. The RTRL calculates temporal likelihood derivatives by evaluating recursively and storing partial derivatives during a forward pass through the network. This is an online algorithm that computes the approximate error gradient at every time step.

The differentiation of the instantaneous log-likelihood $\log \mathcal{L}_{i,t} = \log g(y_t, S_t = i|Y_T)$ of an observation in state $S_t = i$ with respect to the hidden-to-output weights v_k, $0 \leq k \leq K$, of the concrete regime-specific RNN network $f_{i,t}$ is carried out using the chain rule as follows (see Appendix):

$$\frac{\partial \log \mathcal{L}_{i,t}}{\partial v_k} = \frac{\partial \log \mathcal{L}_{i,t}}{\partial f_{i,t}} \frac{\partial f_{i,t}}{\partial v_k} = \frac{(y_t - \mu_i) P(S_t = i|Y_T)}{\sigma_i^2} \varphi'_{f_i} c_{k,t} \tag{13}$$

where φ'_{f_i} is the derivative of the hyperbolic tangent function at the output node $\varphi'_{f_i} = (1 - f_{i,t}^2)$.

The RNN output derivatives with respect to the input-to-hidden node weights w_{nd}, $1 \leq n \leq K$, $1 \leq d \leq L + K$, are formulated using a specific variable z_d, $1 \leq d \leq L + K$, which is necessary to distinguish between weights on connections feeding inputs $z_d = y_{t-l}$, $1 \leq d \leq L$, and weights on recurrent connections feeding context signals $z_d = c_{d-L,t-1}$, $L + 1 \leq d \leq L + K$. These temporal output derivatives are obtained in the following way (see Appendix):

$$\frac{\partial f_{i,t}}{\partial w_{nd}} = \varphi'_{f_i} \left(\sum_{k=1}^{K} \left[v_k \frac{\partial c_{k,t}}{\partial w_{nd}} \right] \right) \tag{14}$$

$$\frac{\partial c_{k,t}}{\partial w_{nd}} = \varphi' \left(\sum_{m=1}^{K} \left[w_{km} \frac{\partial c_{m,t-1}}{\partial w_{nd}} \right] + \delta_{kn} z_d \right) \tag{15}$$

where δ_{kn} is the Kroneker delta function: $\delta_{kn} = 1$ if $k = n$ and 0 otherwise, and φ' is the derivative of the hyperbolic tangent at the particular hidden node. The initial derivatives are taken to be zero.

Regime Variances. The equations for the regime-specific variances are found analogously by differentiating the expected total log-likelihood of each regime ($\log \mathcal{L}_i = -\sum_{t=1}^{T} \log \mathcal{L}_{i,t}$). After taking the derivative with respect to the variance σ_i^2 of regime i, setting it to zero, and solving, we arrive at the following equation:

$$\sigma_i^2 = \frac{\sum_{t=2}^{T} (y_t - \mu_i)^2 P(S_t = i|Y_T)}{\sum_{t=2}^{T} P(S_t = i|Y_T)} \tag{16}$$

which accounts for the errors scaled by the smoothed regime probabilities.

4 Forecasting with RS-RNN

The challenge after learning the parameters of the RS-RNN model is to compute the predictive distribution of the future, unseen data [18]. The predictive distribution can be obtained by weighted averaging of the forecasted component densities:

$$p(y_{t+1}|Y_t) = \sum_{i=1}^{2} P(S_{t+1} = i|Y_t) p(y_{t+1}|S_{t+1} = i, Y_t) \tag{17}$$

which can be made after calculating the regime probabilities as follows:

$$P(S_{t+1} = i|Y_t) = \sum_{i=1}^{2} P(S_{t+1} = i|S_t = j) P(S_t = j|Y_t) \tag{18}$$

One-step ahead predictions of the overall mean can be generated using the regime-specific means computed at the outputs of the recurrent neural networks $E\left[y_{t+1}|S_{t+1}=i,Y_t\right]=f_{i,t+1})$, $i=1,2$, by:

$$E\left[y_{t+1}|Y_t\right]=\sum_{i=1}^{2}P(S_{t+1}=i|Y_t)f_{i,t+1} \tag{19}$$

which is actually a superposition of the network means.

5 Empirical Investigations

Empirical investigations with an implementation of the proposed regime-switching recurrent neural network RS-RNN model were carried out to examine: 1) if it can fit well time series data and discover successfully hidden, unobserved regimes; 2) how does its performance compare with related linear and non-linear models for real-world wind time series processing, and 3) whether the obtained predictive distribution has sufficiently good characteristics to motivate its practical usefulness.

5.1 Processing Simulated Series

The ability of the proposed approach to learn effectively was first tested over series generated by a two-regime switching non-linear model with known parameters. Each regime submodel was represented by a recurrent Perceptron network. The Perceptrons were given as inputs a constant, an autoregressive variable (the most recent value from the series data with lag=1), and a simulated error term e_t which is a random variable $e_t\sim\mathcal{N}(0,\sigma^2_{S_t})$. When training this error term is actually computed via the recurrent feedback connection feeding the past network output. Overall we made an autoregressive Markov regime-switching recurrent Perceptron model $RS(2)$-$RP(1)$ defined as follows:

$$\begin{aligned}
y_t &= \mu_{S_t}+\varepsilon_t,\varepsilon_t\sim\mathcal{N}(0,\sigma^2_{S_t}) && (20)\\
x_t &= [1,e_{t-1},y_{t-1}],e_t\sim\mathcal{N}(0,\sigma^2_{S_t})\\
\mu_{S_t} &= f_2(x_t)S_t+f_3(x_t)(1-S_t),S_t=\{1,2\}\\
\sigma^2_{S_t} &= \sigma^2_1(x_t)S_t+\sigma^2_2(x_t)(1-S_t)\\
f_1(x_t) &= \varphi(w_1x_t),w_1=[w_{11},w_{12},w_{13}] && (21)\\
f_2(x_t) &= \varphi(w_2x_t),w_2=[w_{21},w_{22},w_{23}] && (22)
\end{aligned}$$

where the noise variances were $\sigma^2_1=0.1$ and $\sigma^2_2=0.2$, and the regime weight vectors were $w_1=[0.2,0.1,0.8]$, and $w_2=[-0.2,0.05,0.6]$. With these parameters the first regime features higher autoregressive term, lower recurrent term and lower noise error variance. Note that since there are only two states we have labeled them for clarity with 0 and 1, that is $S_t=i(i=1,2)$. The transition probabilities were $r_{11}=0.9$ and $r_{22}=0.8$.

Using this switching model there were generated two groups of series with length respectively: $T=500$, and $T=1000$. From each length we made 1000 replicas using sampling. Training was carried out starting with sensible initial weight values for the recurrent Perceptron selected as follows: $w_{11} = 0.05$, $w_{12} = 0.05$, $w_{13} = 0.2$, and $w_{21} = -0.05$, $w_{22} = 0.05$, $w_{23} = 0.1$. The sensible weights were computer after conducting a large number of preliminary random restarts by generating random weights from the interval $[0, 1]$.

The $RS(2)$-$RP(1)$ is a nonlinear model that was trained with the Extended Kalman Filter (EKF) [18,19] implemented using the online RTRL derivatives obtained here especially for switching regime recurrent networks (Eq. (13),(14),(15)). The EKF helps us to achieve stable and fast convergence of the neural network training process. The filter was initialized with state variance 1.0e-3 and output noise variance 1.0. The diagonal values of the covariance matrix were set to 1.0e1.

Table 1 gives the estimated average parameters and their Root Mean Squared Errors (RMSE) in parentheses. The values in Table 1 show that the learning algorithm finds adequate parameters, although the precision varies with the sample sizes. These results indicate that the proposed approach has good learning capacity to converge to the true parameters. Another confirmation of its potential is offered by Figures 2 and 3. Figure 2 offers a plot of the fitting one particular sample time series which shows that the curve produced by the concrete learned switching recurrent Perceptron $RS(2)$-$RP(1)$ model is really close to the generated true series. Figure 3 shows that the evolution of the two regimes follows closely the changes in the given true regime.

5.2 Processing Wind Time Series

A time series of wind speed measurements in ms^{-1}, part of the ERA-40 atmospheric data archive, was taken for experiments. The measurements have been recorded at a wind station located in Iceland. These data are freely available from the European Centre for Medium-Range Weather Forecasts (ECMWF), at: `http://apps.ecmwf.int/datasets/data/era40-daily/levtype=sfc`. The series contains data taken at every 6 hours in years 2000 and 2001, there were no missing data and attempts to remove outliers were not made. The whole series consists of 2928 points, which were divided into 1464 for training and 1464 for testing (this splitting was made to facilitate comparisons with similar research [1]). The midday values recorded daily at 12:00 were used as targets, and previous values were taken to form input vectors[1]. The data were scaled to the interval $[0, 1]$ to enable neural network learning.

There were conducted experiments initially with autoregressive (AR) models to identify the proper lagged input dimension, following recent developments on similar wind series [1]. Model selection was performed using the Bayes Information Criterion (BIC): $BIC = -2 \log \mathcal{L} + \Theta \log T$, where Θ is the number of model parameters θ. Starting with $L = 1$, and increasing the inputs up to $L = 10$, we found that optimal fit is achieved using the three most recent lagged values in the series $L = 3$. That is why, in all of the following empirical investigations we considered input vectors $x_t = [1, y_{t-1}, y_{t-2}, y_{t-3}]$.

[1]This research considers univariate wind speed time series, and does not use additional atmospheric data which are problematic to handle and require additional expertise for achieving more accurate results [16].

5.2.1 Studied Models and Algorithms

Reference Models. First we made a linear autoregressive model of order three $AR(3)$. Second, we implemented a Markov Switching linear Autoregressive Model (MS-AR) and a Smooth Transition Autoregressive (STAR) model in order to make our results comparable to recent research [1,34,35]. The STAR was taken as a mixture model that showed better accuracy than the alternative SETAR [35]. A version $STAR(2,3)$ was made with 2 regimes each having a linear autoregressive model of order 3. The logistic function was used to control the transitions between the regimes with slope parameter $\gamma = 1.0$, and midpoint parameter $c = 0.25$. The switching $RS(2)$-$AR(3)$ was elaborated assuming that the real cyclonic conditions lead to two different types of wind: one with high wind speed, and another with low wind speed.

Next, we designed a non-linear feedforward Multilayer Perceptron $MLP(3)$ neural network using 3 lagged inputs (as well as bias terms) and 5 hidden nodes, selected after conducting preliminary model selection with the BIC criterion. Since the MLP is a static model, we also made a dynamic recurrent network $RNN(3)$ [11] that reacts not only to the inputs but also to its temporal context. Finally, an $RS(2)$-$RNN(3)$ model with two regimes for high and low wind speed was built as proposed in this paper. The $RNN(3)$ and $RS(2)$-$RNN(3)$ networks were constructed with 5 hidden nodes.

Estimation Algorithms. The linear submodels and the non-linear neural networks were treated using Kalman (KF) and Extended Kalman Filters (EKF) [18]. Each run of the filter included 10 presentations of the entire time series. The initial weight values were randomly generated from the interval $(-0.5, 0.5)$. All filters were initialized with the same plausible values: state variance $1.0e$-3 and output noise variance 1.0. The diagonal values of the covariance matrix were set to $1.0e1$ [19].

The derivatives for EKF training of the $MLP(3)$ model were obtained using the backpropagation algorithm [36]. The derivatives for EKF training of the $RNN(3)$ and $RS(2)$-$RNN(3)$ were calculated using the RTRL algorithm [46]. More precisely, the $RS(2)$-$RNN(3)$ model was implemented using the novel online RTRL derivatives obtained here especially for switching regime recurrent networks (Eq. (13),(14),(15)).

5.2.2 Experimental Technology and Measures

The training (in-sample) and testing (out-of-sample) performance of the studied models was examined using rolling regression. The wind series was split into training and testing subseries. The training series was used to infer the model parameters. The out-of-sample accuracy was examined by computing one-step-ahead forecasts, and rolling sequentially by one-step foreword over the testing subseries. After predicting the future points the model was retrained, and this algorithm repeated till the end of the testing series. Note that due to the rolling procedure the parameters continuously stabilize because of the repeated retraining over the future series data, and it is reasonable to obtained lower testing errors.

First, we calculated recommended measures for in-sample and out-of-sample evaluation of wind models [28], namely the Mean Squared Error (MSE), the Mean Absolute Error (MAE), the Normalized Mean Squared Error ($NMSE$), the Standard Deviation of Errors (SDE), and the coefficient of determination

(R^2). Second, we carried out statistical diagnostics using standardized residuals $\hat{\epsilon}_t = (y_t - \hat{\mu}_{S_t})/\hat{\sigma}_{S_t}$ in order to assess their deviation from normality. We computed the coefficients of skewness and kurtosis, the Durbin-Watson (D-W) and the Box-Pierce (Portmanteau) (B-P) autocorrelation statistics. The results provided in Tables 2, 3 and 4 below offer diagnostic results computed over the wind time series normalized to the interval $[0, 1]$.

5.2.3 Experimental Results

Tables 2 and 3 provide results from fitting the training and predicting the testing wind speed subseries (the reported values in these tables are calculated over the normalized data illustrated in Figure 5). One can see that the regime-switching $RS(2)$-$AR(3)$ network model has lowest MSE, MAE, $NMSE$ and SDE errors on both the training and the testing subseries. Overall, we found that $RS(2)$-RNN(3 achieves improvements in accuracy as follows: it is 16% better than $AR(3)$, 4.5% better than $STAR(2, 3)$, 13% better than $MS(2)$-$AR(3)$, 7% better than $MLP(3)$ and 4% better than $RNN(3)$. The values of the R^2 coefficient from the $RS(2)$-$RNN(3)$ model are closer to 1 which means that it explains better than the other competing models the variance in the data.

The results from the neural networks given in Tables 2 and 3 indicate that the recurrent networks achieve lower errors compared to feedforward networks. The reason for this is that the recurrent networks have richer potential for learning from dynamic data. This is the explicit handling of the temporal dimension that allows us accurately to model the dynamic evolution of the fluctuations in time series, which confirms the theoretical suggestions [11,19] and the practical findings from relevant research [3,7,33,44].

Table 4 reports the results from statistical diagnostics with the standardized residuals from the models. It shows that all studied approaches lead to models with close skewness and kurtosis. All models have similar values of the Durbin-Watson statistics close to 2.0, which means that there is no significant autocorrelation in their residuals, and this finding is confirmed by the Box-Pierce tests. The nonlinear $RS(2)$-$RNN(3)$ mixture model demonstrates highest likelihood. Overall, however, the nonlinear neural network models including the proposed $RS(2)$-$RNN(3)$ are comparable but not better than the linear $AR(3)$, $STAR(2, 3)$ and $MS(2)$-$AR(3)$ models with respect to these statistical characteristics.

Figure 4 offers a plot of the wind series and its approximation by the studied models. It shows that $RS(2)$-$RNN(3)$ fits better the given series than the competing linear autoregressive ($AR(3)$, switching $STAR(2, 3)$ and $MS(2)$-$AR(3)$) models, as well as the nonlinear $MLP(3)$ and $RNN(3)$ neural models. One can see that the $RS(2)$-$RNN(3)$ curve approximates more precisely the directional changes in the series.

Figure 5.a illustrates the curves of all observations, the mixture $RS(2)$-$RNN(3)$ network output computed by $\hat{\mu}_t = \sum_{i=1}^{R} P(S_{t+1} = i|Y_t) f_{i,t+1}$, and the piecewise line of the maximally probable regime at each time step. The curves in Figure 5.a demonstrate that the output values from the mixture network wrap closely the observations. The computed maximally probable regimes indicate changes in the wind speed that can be explained as follows: from high wind speed in January and February to low wind speed in March, next again to high wind speed in April, next to low wind speed in May and June, as well as in July and August, and finally high wind speed till the end of the year.

Figure 5.b presents a plot of the probability distribution of the two regimes outputs, which are generated by the corresponding recurrent networks, estimated using the transition matrix. It can be seen that actually the means of the outputs from the regime networks are very close. Figure 5.c gives a quantile-quantile (Q-Q) plot of the standardized residuals, which indicates that there is no significant deviation from normality in them. Figure 5.d presents a correlogram of the standardized residuals. The correlogram of the residuals shows that the autocorrelation in the residuals up to 30 lags is small and close to zero.

5.3 Evaluating the Predictive Distribution

When analyzing mixture models, like $RS(2)$-$AR(3)$ and $RS(2)$-$RNN(3)$, one has to judge their usefulness through evaluation of the density forecasts, since they give more precise quantification of the model adequacy compared to point forecasts especially when learning from noisy time series. We use the probability integral transform (PIT) [9,37] which is a general measure for estimating forecasts that can be computed with the empirical cdf of the variable of interest (here y_{t+1}) without relying on distributional assumptions. The idea is to produce a sequence of density forecasts $\left\{\hat{P}(y_t|x_t)_{t=1}^T\right\}$ and relate it to the true but unknown sequence $\left\{\hat{P}(y_t|x_t)_{t=1}^T\right\}$. When the generated density forecasts coincide with the true density the values of the PIT are i.i.d uniform, that is:

$$\{Z_t\}_{t=T+1}^{T+h} = \left\{\int_{-\infty}^{y_t} \hat{P}(u|C_t)du\right\}_{t=T+1}^{T+h} \sim_{i.i.d.} U(0,1) \tag{23}$$

where h is the forecast horizon, $x_t \subseteq C_t$, and y_t are one-step ahead forecasts (actual realizations). That is, the PIT is actually the value of the cdf at the forecast.

5.3.1 Bootstrap Sampling

The effectiveness of the $RS(2)$-$RNN(3)$ network model on forecasting was investigated using bootstrap sampling. We followed a bootstrapping methodology especially for nonlinear neural network models [41]. After fitting the network model to the given training subseries we obtained parameter estimates $\hat{\theta}$ and used them to compute residuals $\hat{\varepsilon}_t = y_t - \hat{\mu}_{S_t}(\hat{\theta})$. Next, samples $\hat{\varepsilon}_t^*$ from these residuals $\hat{\varepsilon}_t$ were drawn, in order to produce replicates of the observations $\{y_1^*, y_2^*, ..., y_T^*\}$ with the equation: $y_t^* = \hat{\mu}_{S_t} + \hat{\varepsilon}_t^*, 1 \leq t \leq T$. After that, the network model was re-estimated over the replicated series leading to a new set of adapted parameters $\hat{\theta}^*$. Thus, we generated 1000 bootstrap copies of the $RS(2)$-$RNN(3)$ model with different parameters. One-step ahead forecasts were produced from each of these 1000 models in order to study the model predictive performance. Since we operate on time series, this bootstrap methodology was implemented using block resampling with blocks of preselected size 10. Figure 6 demonstrates the 95% prediction intervals computed using bootstrapped one-step ahead forecasts from the $RS(2)$-$RNN(3)$ model.

5.3.2 Model-checking Analysis

The interpretation of the bootstrapped one-step ahead forecasts is made using a histogram of the computed PIT values, followed by calculating some additional statistics for uniformity and independence. The PIT measures whether the forecasted cdfs at every future time step are uniformly distributed. In other words, it tells us that if the forecasted cdfs are a random sample $U(0,1)$ and i.i.d. then the model is correct.

Figure 7 gives a histogram of the calculated PIT values over the testing wind speed time series recorded in year 2001. This histogram shows that the density forecasts from both studied models are good, because they are approximately uniformly distributed between 0 and 1, since there are no significant deviations in the different regions. One can observe that the deviations in the different regions in the histogram in figure 7.c are smaller than these in figure 7.a which indicates that the nonlinear $RS(2)$-$RNN(3)$ model is better in predicting the density than the linear $RS(2)$-$AR(3)$ model.

The correlograms of the $(Z - mean(Z))$ series of PIT values from the investigated models plotted in figures 7.b and 7.d provide a support for their independence. One can see that there are no significant autocorrelations in the $(Z - mean(Z))$ series from the $RS(2)$-$RNN(3)$ model, which suggests that these PIT values are independent identically distributed. It should be noted that there are some short-term autocorrelations in the PIT values from the $RS(2)$-$AR(3)$ model.

6 Discussion

The presented work confirmed findings from similar studies that recurrent RNN models can generate highly accurate predictions on wind time series [7]. Although the proposed enhancement with switching regimes used an RNN architecture with hidden-to-hidden node feedback connections (Elman recurrencies), it can be taken to improve Zipser networks with output-to-hidden recurrencies, because they have the same representation potential [46]. The developed switching formalism (Sections 2 and 3) can be made also to use feedforward MLP networks [10,25]. In both cases one simply has to replace the node activations (Eq.(5)), the equations for the regime means (Eq.(13)) and the derivatives (Eq.(24),(25),(26)) with the ones from the corresponding (feedback or feedforward) backpropagation training algorithm [36,46]. Our idea to use nonlinearities through connectionist modeling allows also to modify and upgrade linear regime-switching models [1,13,34,38] by accommodating nonlinear neural network regime submodels.

The suggested switching mechanism can be implemented to operate on data coming from multi-regime sources by making several regimes ($S_t = \{1, ..., R\}$) according to the given formulae (sections 2.1,3.1 and 3.2). Such ensembles can eventually improve further the results because having multi-mixtures allows us to describe more complex density shapes, but this depends on the characteristics of the given data. An arbitrary increase of the number of regimes may not improve much the results on some data [29]. The number of regimes can be determined with a preselection procedure using bootstrap sampling and the PIT measure given here (Section 5.3), or using ranked scoring [16] as in previous research [30].

7 Conclusion

This paper presented an original regime-switching recurrent network (RS-RNN) and related it to some popular techniques for wind time series modeling. It was found that the RS-RNN model: 1) fits accurately the data and features lower residuals than all other models on the training and the testing subseries; 2) attains statistical characteristics (of the standardized residuals) which are better than these of the linear models; 3) achieves more accurate density forecasting as it approximates better the distribution of the data. This allows us also to obtain plausible confidence intervals that quantify the model uncertainty. The overall conclusion is that RS-RNN provides a framework for probabilistic density modeling especially convenient for describing non-stationary time series.

The RS-RNN offers a useful alternative not only for wind speed modeling as demonstrated in this paper, but also for modeling various non-stationary real-world time series as well (such as environmental series, medical series, etc.). Future research will use the regime-switching recurrent network to learn the stochastic behavior of temperatures for valuation of weather derivatives.

Acknowledgements

The authors thank Pierre Ailliot from The University of Brest, France, for providing the preprocessed wind time series data. We also thank Dr Peter Tino from The University of Birmingham for the helpful discussions on dynamic training of recurrent neural networks.

References

[1] Ailliot, P. and Monbet, V. (2012). Markov-Switching Autoregressive Models for Wind Time Series, *Environmental Modelling and Software*, vol.30, pp.92-101.

[2] Barber, D and Cemgil, A.T. (2010). Graphical Models for Time Series, *IEEE Signal Processing Magazine*, vol.27, N:6, pp.18-28.

[3] Barbounis, T.G. and Theocharis, J.B. (2007). A Locally Recurrent Fuzzy Neural Network with Application to the Wind Speed Prediction using Spatial Correlation, *Neurocomputing*, vol.70, pp.1525-1542.

[4] Bessac, J., Ailliot, P., Cattiaux, J. and Monbet, V. (2016). Comparison of Hidden and Observed Regime-switching Autoregressive Models for (u,v)-components of Wind Fields in the Northeastern Atlantic, *Advances in Statistical Climatology, Meteorology and Oceanography*, vol.2, pp.1-16.

[5] Brka,A., Al-Abdeli,Y.M. and Kothapalli,G. (2016) Influence of neural network training parameters on short-term wind forecasting, *Int. Journal of Sustainable Energy*, vol.35, pp.115-131.

[6] Browell, J., Drew, D.R. and Philippopoulos, K. (2018). Improved very short-term spatio-temporal wind forecasting using atmospheric regimes, *Wind Energy*.

[7] Cao, Q., Ewing, B.T. and Thompson, M.A. (2012). Forecasting Wind Speed with Recurrent Neural Networks, *European Journal of Operations Research*, vol.221, pp.148-154.

[8] Chang,W.-Y. (2014) A Literature Review of Wind Forecasting Methods, *Journal of Power and Energy Engineering*, vol.2, pp.161-168.

[9] Diebold, F.X., Gunther, T.A. and Tay, A.S. (1998). Evaluating Density Forecasts with Applications to Risk Management, *International Economic Review*, vol.39, pp.863-883.

[10] do Nascimento Camelo, H., Lucio, S., Leal Junior, J.B, dos Santos, G., de Carvalho, P.C.M. (2018). Innovative Hybrid Modeling of Wind Speed Prediction Involving Time-Series Models and Artificial Neural Networks, *Atmosphere*, vol.9, N:2.

[11] Elman, J.J. (1990). Finding Structure in Time, *Cognitive Science*, vol.14, pp.179-211.

[12] Evarest,E., Berntsson,F., Singull,M. and Charles,W.M. (2017). Regime Switching models on Temperature Dynamics, *Int.l Journal of Applied Mathematics and Statistics*, vol.56, N:2, pp.19-36.

[13] Gel, Y.R., Lyubchich, V. and Ahmed, S.E. (2016). Catching Uncertainty of Wind: A Blend of Sieve Bootstrap and Regime Switching Models for Probabilistic Short-Term Forecasting of Wind Speed, In: *Advances in Time Series Methods and Applications*, Springer, New York, pp.279-293.

[14] Giebel, G. and Kariniotakis, G. (2017). Wind power forecasting a review of the state of the art In: *Renewable Energy Forecasting: From Models to Applications*, Elsevier- Woodhead Publ. Series in Energy, Chapter 3, pp.59-96.

[15] Gnana Sheela, K. and Deepa, S.N. (2013). Neural Network Based Hybrid Computing Model for Wind Speed Prediction, *Neurocomputing* , vol.122, pp.425-429.

[16] Gneiting, T., Larson, K., Westrick, K, Genton, M.G. and Aldrich, E. (2006). Calibrated probabilistic forecasting at the Stateline Wind Energy Center: The Regime-switching Space-time Method, *Journal of the American Statistical Association*, vol.101, pp.968-979.

[17] Guo, Z.H., Wu,J., Lu,H.Y. and Wang,J.Z. (2011). A Case Study on a Hybrid Wind Speed Forecasting Method using BP Neural Network, *Knowledge-Based systems*, vol.24, pp.1048-1056.

[18] Hamilton, J.D. (1994). *Time Series Analysis*, Princeton University Press, NJ.

[19] Haykin, S. (Ed.) (2000). *Kalman Filtering and Neural Networks*, John Wiley and Sons, New York.

[20] Hering, A.S., Kazor, K. and Kleiber, W. (2015). A Markov-Switching Vector Autoregressive Stochastic Wind Generator for Multiple Spatial and Temporal Scales, *Resources*, vol.4, N:1, pp.70-92.

[21] Jung, J. and Broadwater, R.P. (2014). Current Status and Future Advances for Wind Speed and Power Forecasting, *Journal of Renewable and Sustainable Energy Resources*, vol.31, pp.762-777.

[22] Kim, C.J. (1994). Dynamic Linear Models with Markov Switching, *Journal of Econometrics*, vol.60, pp.1-22.

[23] Koskela, T., Lehtokangas, M., Saarinen, J. and Kaski, K. (1996). Time Series Prediction with Multilayer Perceptron, FIR and Elman Neural Networks, In: Proc. WCNN-96, pp.491-496.

[24] Lau, A. and McSharry, P. (2010). Approaches to Multi-step Density Forecasts with Application to Aggregated Wind Power, *Annals of Applied Statistics*, vol.4, pp.1311-1341.

[25] Li, G. and Shi, J. (2010). On Comparing Three Artificial Neural Networks for Wind Speed Forecasting, *Applied Energy*, vol.87, N:7, pp.2313-2320.

[26] Li, J., Zhang, B., Mao, C-H., Xie, G., Lie, Y. and Lu, J. (2010). Wind Speed Prediction based on Elman Recursion Neural Networks, In: Proc. Int. Conf. on Modelling, Identification and Control, ICMIC-2010, pp.728-732.

[27] Lydia, M., Suresh Kumar, S., Selvakumar, A.I. and Kumar, G.E.P. (2018). Linear and non-linear autoregressive models for short-term wind speed forecasting *Energy Conversion and Management*, vol.112, pp.115-124.

[28] Madsen, H., Pinson, P., Kariniotakis, G., Nielsen, H.A., and Nielsen, T.S. (2005). Standardizing the Performance Evaluation of Short-term Wind Power Prediction Models, *Wind Engineering*, vol.29, N:6, pp.475-489.

[29] McLachlan, G.J. and Peel, D. (2000). *Finite Mixture Models*, John Wiley and Sons, New York.

[30] Men,Z, Yee,E., Lien,F.-S., Wen,D. and Chen,Y. (2016). Short-term Wind Speed and Power Forecasting using an Ensemble of Mixture Density Neural Networks, *Renewable Energy*, vol.87, pp.203-211.

[31] Monbet, V., Ailliot, P. and Prevosto, M. (2007). Survey of Stochastic Models for Wind and Sea State Time Series, *Probabilistic Engineering Mechanics*, vol.22, pp.113-126.

[32] Monbet, V. and Ailliot P. (2017). Sparse Vector Markov Switching Autoregressive Models. Application to Multiple Time Series of Air Temperature, *Computational Statistics and Data Analysis*, vol.108, pp.40-51.

[33] More,A. and Deo,M.C. (2003) Forecasting Wind with Neural Networks. *Marine Structures*, vol.16, pp.35-49.

[34] Pinson, P. and Madsen, H. (2012). Adaptive Modelling and Forecasting of Offshore Wind Power Fluctuations with Markov-Switching Autoregressive Models, *Journal of Forecasting*, vol.31, N:4, pp.281-313.

[35] Pinson, P., Christensen, L.E.A., Madsen, H., Sørensen, P.E., Donovan, M.H. and Jensen, L.E. (2008). Regime-switching Modelling of the Fluctuations of Offshore Wind Generation, *Journal of Wind Engineering and Industrial Aerodynamics*, vol.96, N:12, pp.2327-2347.

[36] Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986). Learning Internal Representations by Error Propagation. In: D.E.Rumelhart and J.L.McLelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, The MIT Press, Cambdidge, MA, vol.1, ch.8, pp.318-362.

[37] Shi, S. and Weigend, A.S. (1997). Taking Time Seriously: Hidden Markov Experts Applied to Financial Engineering, In: Proc. IEEE/IAFE Conf. Comput. Intelligence for Financial Engineering (CIFEr), New York, pp.244-252.

[38] Song, Z., Jiang, Y. and Zhang, Z. (2014). Short-term Wind Speed Forecasting with Markov-Switching Model, *Applied Energy*, vol.130, pp.103-112.

[39] Spezia, L. and Paroli, R. (2008). Bayesian Inference and Forecasting in Dynamic Neural Networks with Fully Markov Switching ARCH Noises, *Communications in Statistics- Theory and Methods*, vol.37, pp.2079-2094.

[40] Taylor, J.W., McSharry, P. and Buizza, P.E. (2009). Wind Power Density Forecasting using Wind Ensemble Predictions and Time Series Models, *IEEE Trans. on Energy Conversion*, vol.24, pp.775-782.

[41] Tibshirani, R. (1996). A Comparison of Some Error Estimates for Neural Network Models, *Neural Computation*, vol.8, N:1, pp.152-163.

[42] Vasas, K., Elek, P. and Markus, L. (2007). A Two-state Regime Switching Model with An Application to River Flow Analysis, *Journal of Statistical Planning and Inference*, vol.137, pp.3113-3126.

[43] Velo, R., Lpez, P. and Maseda, F. (2014). Wind Speed Estimation using Multilayer Perceptron, *Energy Conversion and Management*, vol.81, pp.1-9.

[44] Wang, J., Zhang, W., Li, Y., Wang, J. and Dang, Z. (2014). Forecasting Wind Speed using Empirical Mode Decomposition and Elman Neural Network, *Journal of Applied Soft Computing*, vol.23, pp.452-459.

[45] Weigend, A.S. and Shi, S. (2000). Predicting Daily Probability Distributions of SandP500 Returns, *Journal of Forecasting*, vol.19, N:4, pp.375-392.

[46] Williams, R.J. and Zipser, D. (1989). A Learning Algorithm for Continually Running Fully Recurrent Networks, *Neural Computation*, vol.1, pp.270-280.

[47] Zhang, Y., Wang, J. and Wang, X. (2014). Review on Probabilitics Forecasting of Wind Power Generation, *Renewable and Sustainable Energy Reviews*, vol.32, pp.255-270.

[48] Zhu, X. and Genton, M.G. (2012). Short-term Wind Speed Forecasting for Power System Operations, *Int. Statistical Review*, vol.80, pp.2-23.

[49] Ziel, F., Croonenbroeck, C. and Ambach, D. (2016). Forecasting wind power Modeling periodic and non-linear effects under conditional heteroscedasticity, *Applied Energy*, vol.177, pp.285-297.

Appendix: Calculation of the RS-RNN Temporal Network Derivatives

Following the RTRL approach [46] to obtain the RS-RNN training rules involves the calculation of the derivatives of the instantaneous log-likelihood with respect to each weight. Let assume that the i-th regime-specific RNN network generates output $f_{i,t} = \varphi_{f_i}(o_{i,t})$, where $o_{i,t} = \sum_{k=1}^{K} v_k c_{k,t}$. The derivative of the instantaneous log-likelihood $\log \mathcal{L}_{i,t} = \log g(y_t, S_t = i|Y_T)$ of an observation in state $S_t = i$ with respect to the hidden-to-output weights v_k, $1 \leq k \leq K$, in this RS-RNN is taken as follows:

$$\begin{aligned}
\frac{\partial \log \mathcal{L}_{i,t}}{\partial v_k} &= \frac{\partial \log \mathcal{L}_{i,t}}{\partial f_{i,t}} \frac{\partial f_{i,t}}{\partial o_{i,t}} \frac{\partial o_{i,t}}{\partial v_k} \\
&= \frac{(y_t - \mu_i) P(S_t = i|Y_{t-1})}{\sigma_i^2} \varphi'_{f_i} \frac{\partial \left[\sum_{k=1}^{K} v_k c_{k,t}\right]}{\partial v_k} \\
&= \frac{(y_t - \mu_i) P(S_t = i|Y_{t-1})}{\sigma_i^2} \varphi'_{f_i} c_{k,t} \qquad (24)
\end{aligned}$$

The derivatives of the output with respect to input-to-hidden weights are taken in two steps:

$$\begin{aligned}
\frac{\partial f_{i,t}}{\partial w_{nd}} &= \frac{\partial f_{i,t}}{\partial o_{i,t}} \frac{\partial o_{i,t}}{\partial w_{nd}} \\
&= \varphi'_{f_i} \frac{\partial \left[\sum_{k=1}^{K} v_k c_{k,t}\right]}{\partial w_{nd}} \\
&= \varphi'_{f_i} \sum_{k=1}^{K} \left[v_k \frac{\partial c_{k,t}}{\partial w_{nd}} + c_{k,t} \frac{\partial v_k}{\partial w_{nd}} \right] \\
&= \varphi'_{f_i} \left(\sum_{k=1}^{K} \left[v_k \frac{\partial c_{k,t}}{\partial w_{nd}} \right] \right) \qquad (25)
\end{aligned}$$

where $\partial v_k / \partial w_{nd} = 0$ because the hidden-to-output weights do not depend on the input-to-hidden weights.

Let assume that the k-th hidden node in the recurrent network produces output $c_{k,t} = \varphi(o_{k,t})$, where $o_{k,t} = \sum_{l=1}^{L} w_{kl} y_{t-l} + \sum_{m=1}^{K} w_{km} c_{m,t-1}$. Then we have:

$$\begin{aligned}
\frac{\partial c_{k,t}}{\partial w_{nd}} &= \frac{\partial c_{k,t}}{\partial o_{k,t}} \frac{\partial o_{k,t}}{\partial w_{nd}} \\
&= \varphi' \frac{\partial \left[\sum_{l=1}^{L} w_{kl} y_{t-l} + \sum_{m=1}^{K} w_{km} c_{m,t-1}\right]}{\partial w_{nd}} \\
&= \varphi' \left(\sum_{l=1}^{L} \left[w_{kl} \frac{\partial y_{t-l}}{\partial w_{nd}} \right] + y_{t-l} \frac{\partial w_{kl}}{\partial w_{nd}} + \sum_{m=1}^{K} \left[w_{km} \frac{\partial c_{m,t-1}}{\partial w_{nd}} \right] + c_{m,t-1} \frac{\partial w_{km}}{\partial w_{nd}} \right) \\
&= \varphi' \left(\sum_{m=1}^{K} \left[w_{km} \frac{\partial c_{m,t-1}}{\partial w_{nd}} \right] + \delta_{kn} y_{t-i} + \delta_{kn} c_{m,t-1} \right) \qquad (26)
\end{aligned}$$

which uses the fact that $\partial y_{t-i} / \partial w_{nd} = 0$ because the inputs do not depend on the weights.

Both Eq. (25) and (26) for the derivatives of the dynamic variables are similar in that their rightmost multipliers in the parentheses contain two similar terms. The first term accounts for the implicit effect of the weight w_{nd} on the hidden node activations $c_{k,t}$ and $c_{m,t-1}$, while the second term is the explicit effect of the weight w_{nd} on the particular n-th network node.

Table 1. Estimated average parameters and their RMSE errors in parentheses obtained after independent runs over 1000 simulated series of increasing sizes T $(500, 1000)$ generated with the chosen true parameters for the switching recurrent Perceptron model $RS(2)$-$RP(1)$.

Parameter	True	$RS(2)$-$RP(1)$	
		$T = 500$	$T = 1000$
w_{01}	0.2	0.1922	0.2065
		(0.0361)	(0.0244)
w_{02}	0.1	0.1176	0.1052
		(0.0278)	(0.0233)
w_{03}	0.8	0.7924	0.8061
		(0.0385)	(0.0396)
σ_0^2	0.1	0.1042	0.0927
		(0.0110)	(0.0156)
r_{00}	0.9	0.8924	0.9071
		(0.0451)	(0.0467)
w_{11}	-0.2	-0.2102	-0.1985
		(0.0285)	(0.0304)
w_{12}	0.05	0.05516	0.0499
		(0.0249)	(0.0258)
w_{13}	0.6	0.5838	0.6023
		(0.0355)	(0.0321)
σ_1^2	0.2	0.2032	0.2154
		(0.0157)	(0.0129)
r_{11}	0.8	0.7856	0.7926
		(0.0445)	(0.0482)

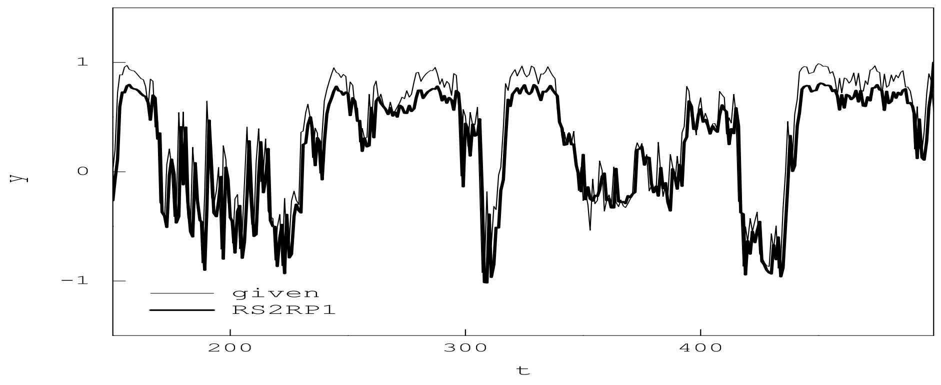

Figure 2. Curves of a segment from a sample simulated time series and its approximation by an estimated switching recurrent Perceptron $RS(2)$-$RP(1)$ network.

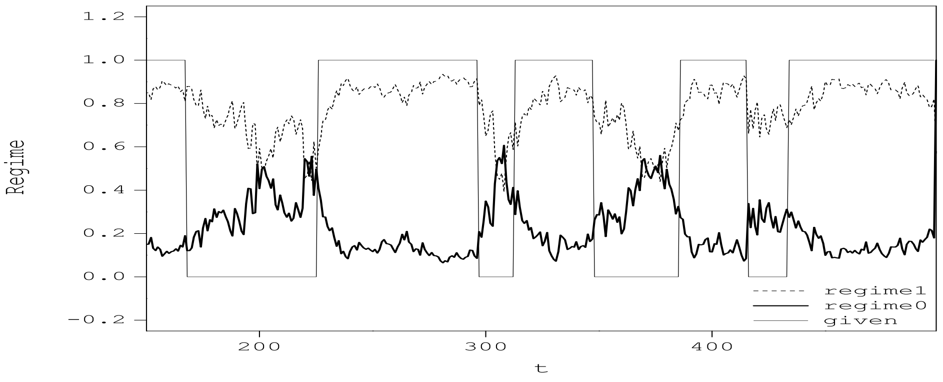

Figure 3. Filtered regime probabilities $P(S_t = i|Y_t)$ (shifts between the two regimes) within a segment from a sample simulated time series given in Figure 2.

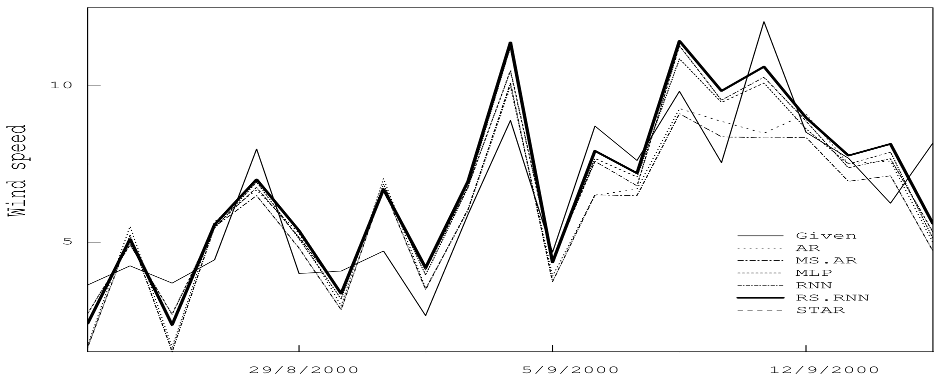

Figure 4. A segment from the given wind speed time series recorded daily at 12:00 in 2000 and its approximation with the studied linear and non-linear models. The $MS(2)$-$AR(3)$ curve is lowest, slightly above is the $STAR(2,3)$ curve, next is the $AR(3)$ curve, and higher are the $MLP(3)$, $RNN(3)$, and $RS(2)$-$RNN(3)$ curves.

Table 2. Training (in-sample) errors of different linear and non-linear neural network models obtained over the training wind speed subseries of 1464 data points recorded in year 2000.

Model	MSE	MAE	NMSE	SDE	R^2
$AR(3)$	3.4592	1.4506	0.3011	1.5814	0.8353
$STAR(2,3)$	3.4669	1.4871	0.2974	1.6355	0.8218
$MS(2)$-$AR(3)$	3.4725	1.5122	0.3126	1.6123	0.8251
$MLP(3)$	3.1242	1.4127	0.2913	1.4486	0.8724
$RNN(3)$	2.9843	1.3862	0.2898	1.4294	0.8729
$RS(2)$-$RNN(3)$	2.8828	1.3514	0.2607	1.3812	0.8729

Table 3. Testing (out-of-sample) errors of different linear and non-linear models computed with one-step-ahead forecasts, obtained via rolling regression over the testing subseries of 1464 future data recorded in year 2001.

Model	MSE	MAE	NMSE	SDE	R^2
$AR(3)$	2.6978	1.2694	0.2487	1.2954	0.8722
$STAR(2,3)$	2.6681	1.2547	0.2462	1.2856	0.8718
$MS(2)$-$AR(3)$	2.6587	1.2496	0.2451	1.2811	0.8723
$MLP(3)$	2.7024	1.2915	0.2512	1.3398	0.8715
$RNN(3)$	2.6209	1.2536	0.2416	1.2972	0.8793
$RS(2)$-$RNN(3)$	2.5603	1.2362	0.2359	1.2654	0.8794

Table 4. In-sample statistical diagnostics calculated with standardised residuals computed after fitting $RS(2)$-$RNN(3)$ to the training wind speed series.

Model	Skewness	Kurtosis	D-W	B-P	Log-lik.
$AR(3)$	0.3588	3.3816	1.9554	0.0617	319
$STAR(2,3)$	0.2631	3.3125	2.0129	0.0753	350
$MS(2)$-$AR(3)$	0.2176	3.2652	2.0437	0.0742	348
$MLP(3)$	0.3427	3.3561	1.9216	0.0758	325
$RNN(3)$	0.3460	3.3627	1.9124	0.0740	351
$RS(2)$-$RNN(3)$	0.3415	3.5714	1.9580	0.0497	364

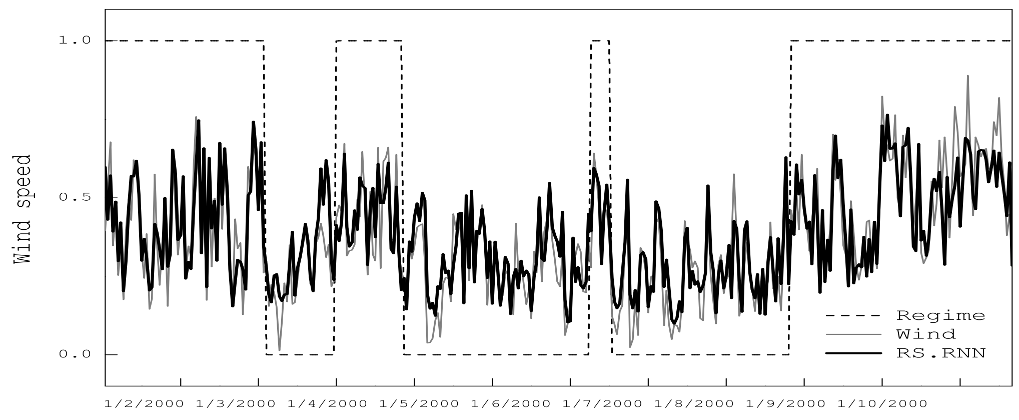

Figure 5.a. Wind speed training series recorded daily at 12:00 from 1 January to 31 December 2000, and their approximation by the mixture $RS(2) - RNN(3)$ model, and changes in the regimes of high speed and low speed (these are values normalized to $[0, 1]$).

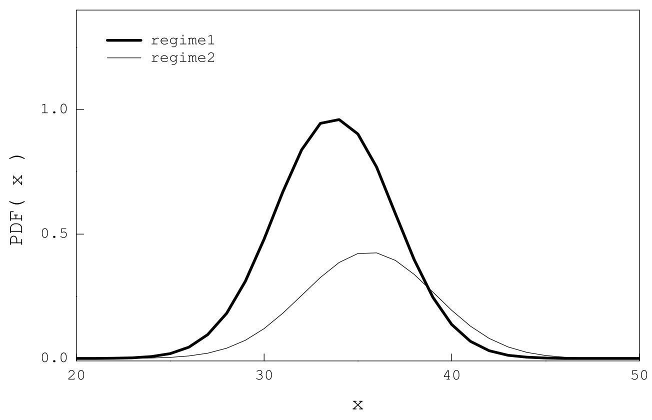

Figure 5.b. Estimated probability distribution of the outputs from the two regime recurrent networks, obtained after learning from the training wind series.

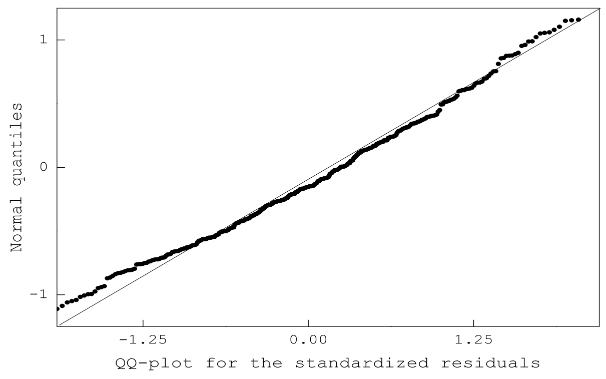

Figure 5.c. Quantile-quantile plot of the standardized residuals from the mixture $RS(2) - RNN(3)$ model, obtained after learning from the training wind series.

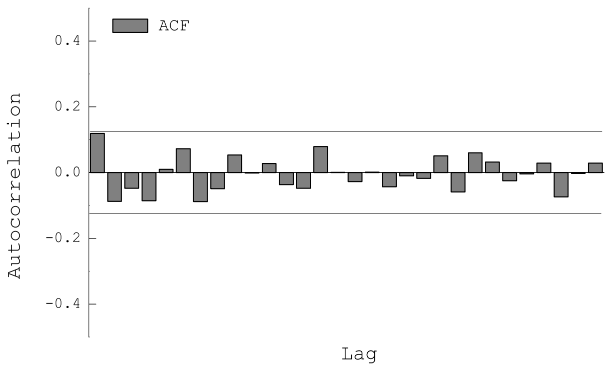

Figure 5.d. Correlogram of the standardized residuals from the mixture $RS(2) - RNN(3)$ model, obtained after learning from the training wind series.

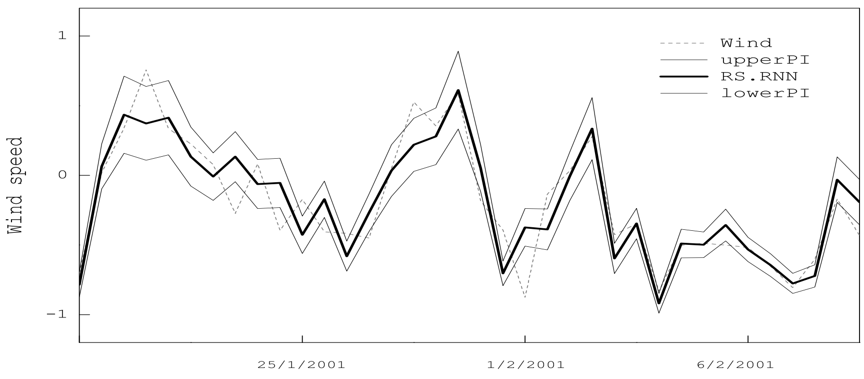

Figure 6. Prediction intervals (95%) of of one-step ahead forecasts obtained by re-estimating the $RS(2)-RNN(3)$ model over 1000 bootsrapped samples of the wind speed time series.

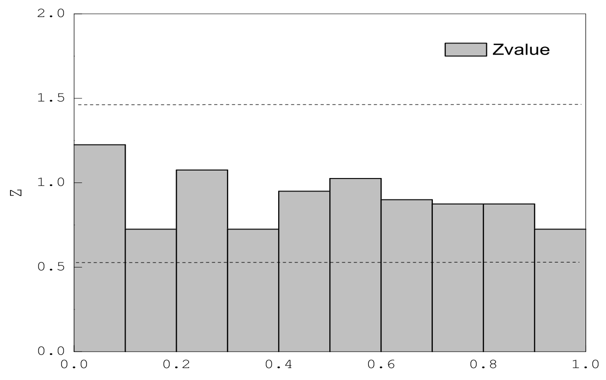

Figure 7.a. Histogram of the Z-values (probability integral transform) calculated with one-step ahead predictions from the $MS(2) - AR(3)$ model obtained via rolling regression over the testing wind speed time series.

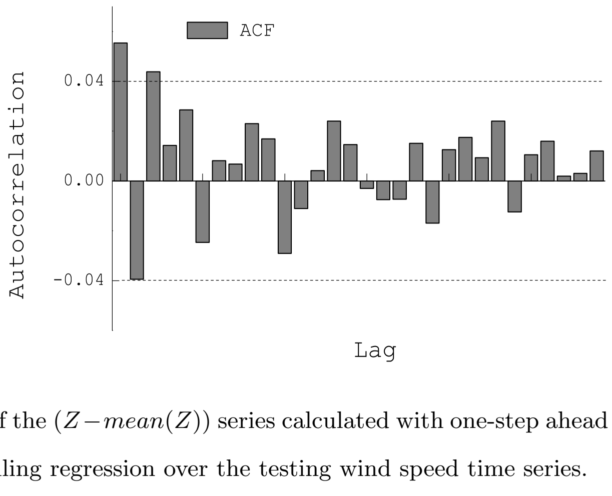

Figure 7.b. Correlogram of the $(Z-mean(Z))$ series calculated with one-step ahead predictions from the $MS(2)-AR(3)$ model obtained via rolling regression over the testing wind speed time series.

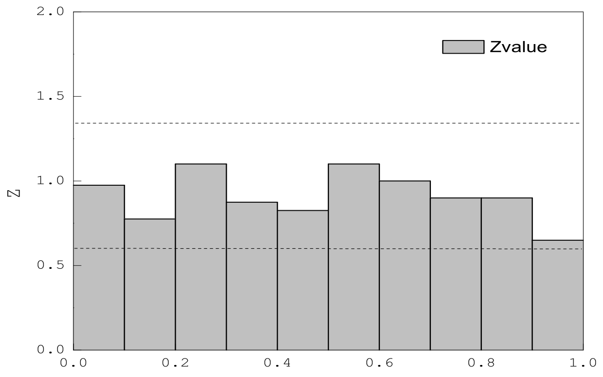

Figure 7.c. Histogram of the Z-values (probability integral transform) calculated with one-step ahead predictions from the $RS(2) - RNN(3)$ model obtained via rolling regression over the testing wind speed time series.

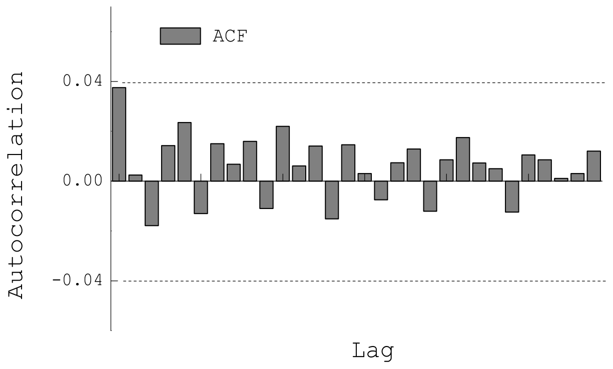

Figure 7.d. Correlogram of the $(Z - mean(Z))$ series calculated with one-step ahead predictions from the $RS(2) - RNN(3)$ model obtained via rolling regression over the testing wind speed time series.