ANL/DIS-11-7

Development and Testing of Improved Statistical Wind Power Forecasting Methods

Decision and Information Sciences Division

ANL/DIS-11-7

Development and Testing of Improved Statistical Wind Power Forecasting Methods

by
Joana Mendes, Ricardo J. Bessa, Hrvoje Keko, Jean Sumaili, Vladimiro Miranda,
Carlos Ferreira, and João Gama
Instituto de Engenharia de Systemas E Computadores do Porto

Audun Botterud, Zhi Zhou, and Jianhui Wang
Decision and Information Sciences Division, Argonne National Laboratory

September 30, 2011

CONTENTS

CONTENTS (CONT.)

FIGURES

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

FIGURES (CONT.)

TABLES

TABLES (CONT.)

ACKNOWLEDGMENTS

This report has been prepared by Argonne National Laboratory in collaboration with INESC Porto, Portugal. The authors acknowledge the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy through its Wind & Water Power Program for funding the research presented in this report under contract DE-AC02-06CH11357.

The authors are grateful to Horizon Wind Energy for providing wind farm data used in the testing of the wind power forecasting algorithms documented in this report. The authors also acknowledge Emil Constaninescu at Argonne National Laboratory for generating the corresponding numerical weather predictions.

Argonne National Laboratory, September 30, 2011.

This page intentionally blank

ACRONYMS

AE	absolute error
Argonne	Argonne National Laboratory
BS	Brier score
CKDE	conditional kernel density estimation
cMCC	centered maximum correntropy criterion
CSI	critical success index
EDS	extreme dependency score
ERCOT	Electric Reliability Council of Texas
EWITS	Eastern Wind Integration and Transmission Study
FN	false negative alarms
FP	false positive alarms
i.i.d	independent and identically distributed (data)
ITL	information theoretic learning
KDE	kernel density estimation
KDF	kernel density forecast
KSS	Hanssen & Kuiper's skill score
MAPE	mean absolute percentage error
MCC	maximum correntropy error
MEE	minimum error entropy
MEEF	minimum error entropy with fiducial points
MSE	minimum square error
NBIAS	normalized bias
NMAE	normalized mean absolute error
NN	neural network
NREL	National Renewable Energy Laboratory
NRMSE	Normalized Root Mean Square Error
NW	Nadaraya-Watson
NWP	numerical weather prediction
OR	odds ratio
pdf	probability density function
pmf	probability mass function
QC	quantile-copula
QR	quantile regression

RMSE	root mean square error
ROC	receiver operating characteristic
SCADA	supervisory control and data acquisition
SDE	Standard Deviation of the Errors
SO	system operator
Std	standard deviation
TN	true negative alarms
TNR	true negative rate
TP	true positive alarms
TPR	true positive rate
W2P	wind-to-power
WFA	wind farm A
WFB	wind farm B
WGENCO	wind generation company
WPF	wind power forecasting
WRF	weather research forecast

EXECUTIVE SUMMARY

In this report, we have documented improved statistical methods for wind power forecasting (WPF). First, we present the results of the application of information theoretic learning (ITL) training criteria to wind power point forecasting. Second, we present novel time-adaptive kernel density forecast (KDF) methods for characterizing WPF uncertainty, along with the corresponding case study results. Finally, a new method to predict and visualize ramp events is illustrated. The main conclusions and contributions to the current state-of-the-art for each area of research are summarized below.

Point forecasts of wind power are highly dependent on the training criteria used in the underlying statistical algorithms. Our work on *wind power point forecasting* focused on the training criteria used in the computational learning algorithms (we used a neural network), which convert weather forecasts and observational data into a point forecast for wind power. In particular, we used ITL training criteria, which are not built on the assumption of a Gaussian distribution of the forecasting errors.

We applied the ITL training criteria to wind power point forecasts for two wind farms located in the U.S. Midwest. The main findings from our study include the following:

- Demonstration of the advantages of using ITL criteria over the classic Minimum Square Error (MSE) criterion, in terms of reduced forecasting error. Fig. S1 presents an example.
- The improvements of the ITL criteria are particularly significant for low and high wind power output levels.
- A new ITL-based training criterion, centered correntropy, was introduced for the first time in this report.
- Among several ITL-based criteria, the maximum correntropy criterion (MCC) showed good results and also has a low computational burden.
- The importance of online training, as illustrated in Fig. S2, which assures better results in the presence of concept drift in the training data.

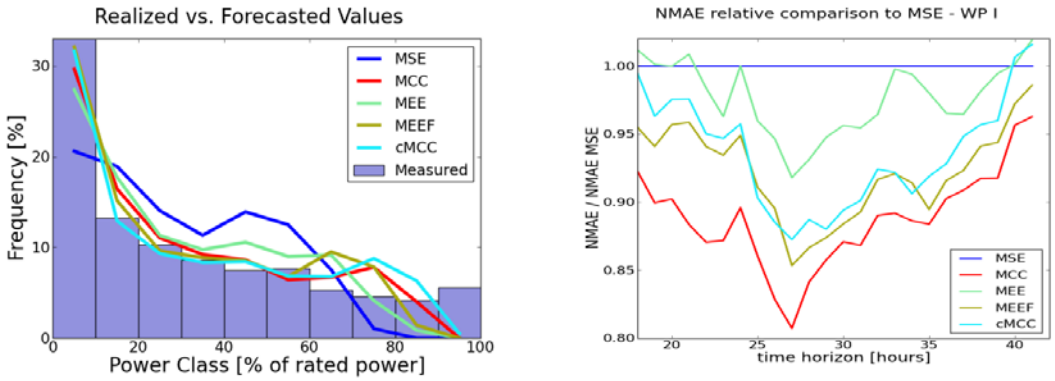

Fig. S1 Improvements in forecast accuracy with different ITL training criteria (MCC, minimum error entropy [MEE], minimum error entropy with fiducial points [MEEF], centered maximum correntropy criterion [cMCC]) compared to traditional MSE for a wind farm in the Midwest. Left: Realized vs. forecasted wind power generation for different power output ranges. Right: Relative improvements in normalized mean average error (NMAE) compared to MSE for different forecast horizons.

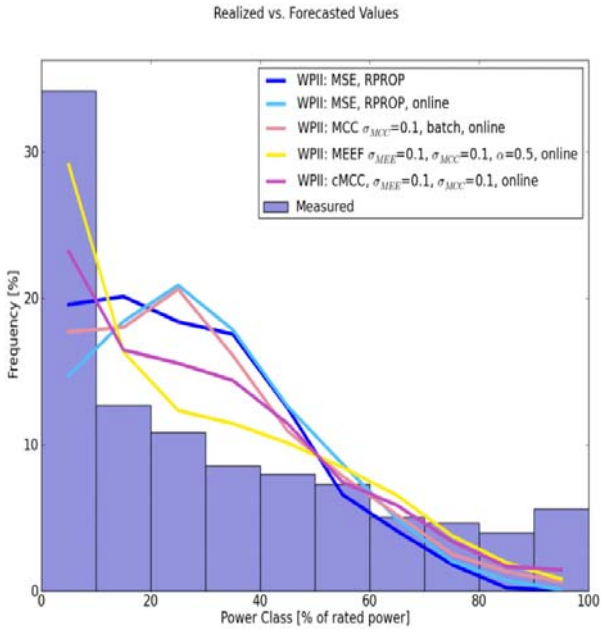

Fig. S2 Comparison of performance of various ITL criteria (MCC, MEE, MEEF, cMCC) with MSE: frequency of occurrence of forecasted and measured values for a Midwest wind farm using online training.

Although there have been advances in deterministic WPF, a single-valued point forecast cannot provide information on the dispersion of observations around the predicted value. Hence, it is essential to generate, together with (or as an alternative to) point forecasts, a representation of the wind power uncertainty. Within *wind power uncertainty forecasting,* we have developed two new probabilistic methods, both based on conditional kernel density estimation. The first method uses the Nadaraya-Watson (NW) estimator, whereas the second method uses the Quantile-

Copula (QC) estimator. We presented time-adaptive versions for both algorithms, which is an important contribution to the current state-of-the-art. We applied the new uncertainty forecasting algorithms in different case studies, comparing the results to linear and splines quantile regression (QR), which are two methods commonly used for statistical estimation of WPF uncertainty.

The new KDF algorithms were tested on a dataset from the Eastern Wind Integration and Transmission Study (EWITS), as well as on two large-scale wind farms located in the U.S. Midwest.

The main achievements of our study include the following:

- The selection of kernels (type, size) is very important for KDF methods. We identified adequate kernels specifically for the WPF problem.
- The new KDF methods tend to yield a better performance than do QR methods in terms of calibration (Fig. S3). QR methods have a tendency to present a better performance in terms of sharpness and resolution (Figs. S4 and S5). KDF and QR methods exhibit similar levels of performance in terms of a skill score (Fig. S6).
- The time-adaptive KDF approach improves the skill score when compared to the offline approach (Fig. S7).
- An important advantage of KDF is that it estimates the full probability distribution for wind power at any forecast horizon.

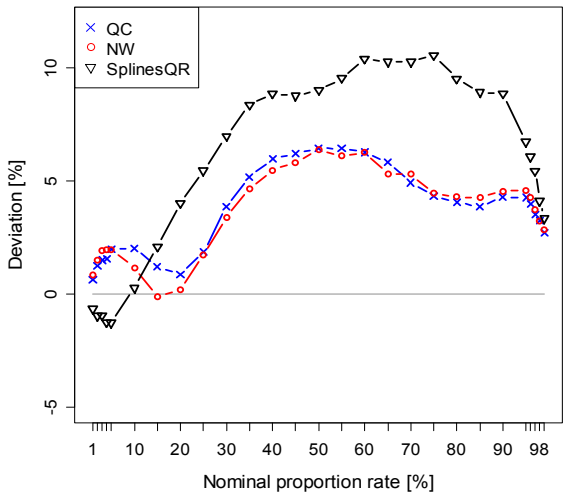

Fig. S3 Calibration diagram for the offline test for EWITS data.

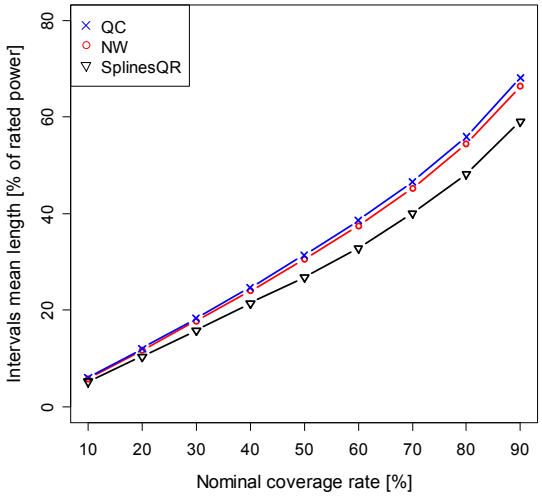

Fig. S4 Sharpness diagram for the offline test for EWITS data.

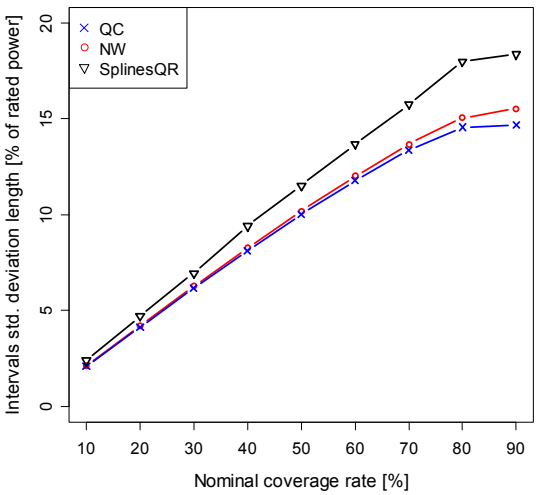

Fig. S5 Resolution diagram for the offline test for EWITS data.

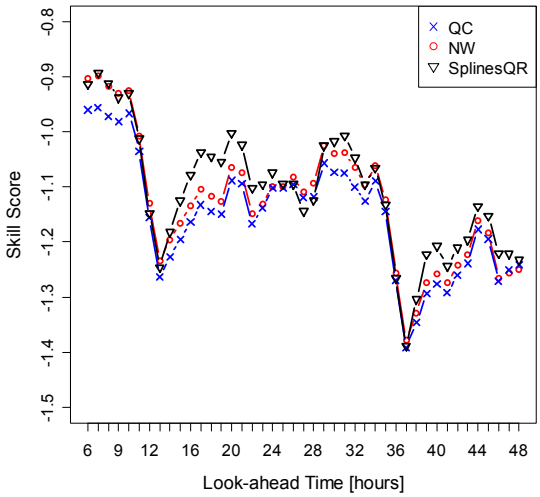

Fig. S6 Skill score diagram for the offline test for EWITS data.

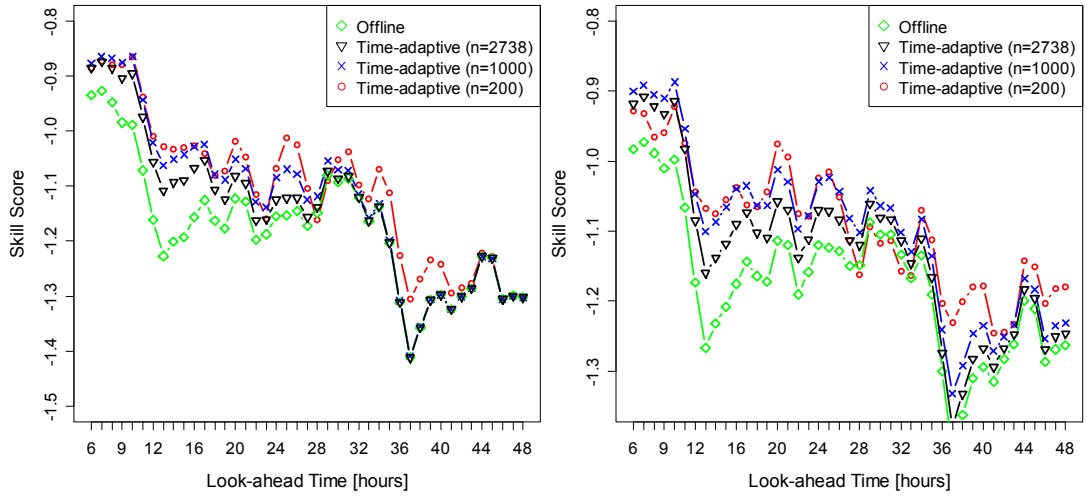

Fig. S7 Skill score diagrams for a Midwest Wind Farm dataset, comparing offline and time-adaptive models. Left: NW; right: QC.

One of the major issues that concern system operators balancing supply and demand in the power grid are sudden and large changes of power output over a short period of time. These are referred to as ramp events, and they can take the form of either an increase or a decrease of wind power generation. For *wind power ramp forecasting*, we have proposed a new method to predict and visualize ramp events based on high-pass filter concepts. The method starts with a large sample of wind power scenarios, which are sampled with a Monte-Carlo approach, from a probabilistic forecast. Scenarios are filtered by using any definition of ramp events. Then, the probability of a ramp event is estimated by using the percentage of scenarios detecting the event, according to its definition. This process is repeated for several ramp magnitudes. The ramp forecasting process is illustrated in Fig. S8, along with a potential application to the stochastic unit commitment problem.

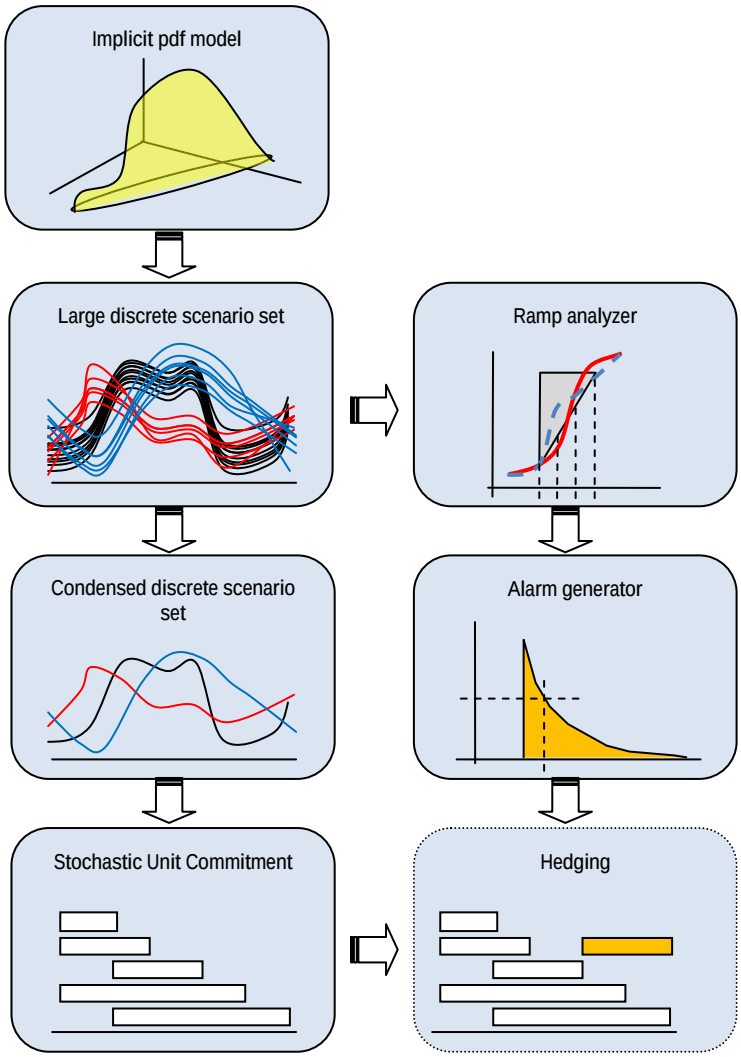

Fig. S8 Conceptual modules relating scenario generation, ramp event analysis, and stochastic unit commitment.

We assessed and validated the behavior and performance of the proposed methodology using experimental data, with different ramping definitions. Ramp uncertainty was represented by the cumulative distribution of ramp probability, within a predefined window, and the visualization of ramp events included histograms of cumulative ramp probability functions and ROC curves, respectively, as illustrated in Figs. S9 and S10.

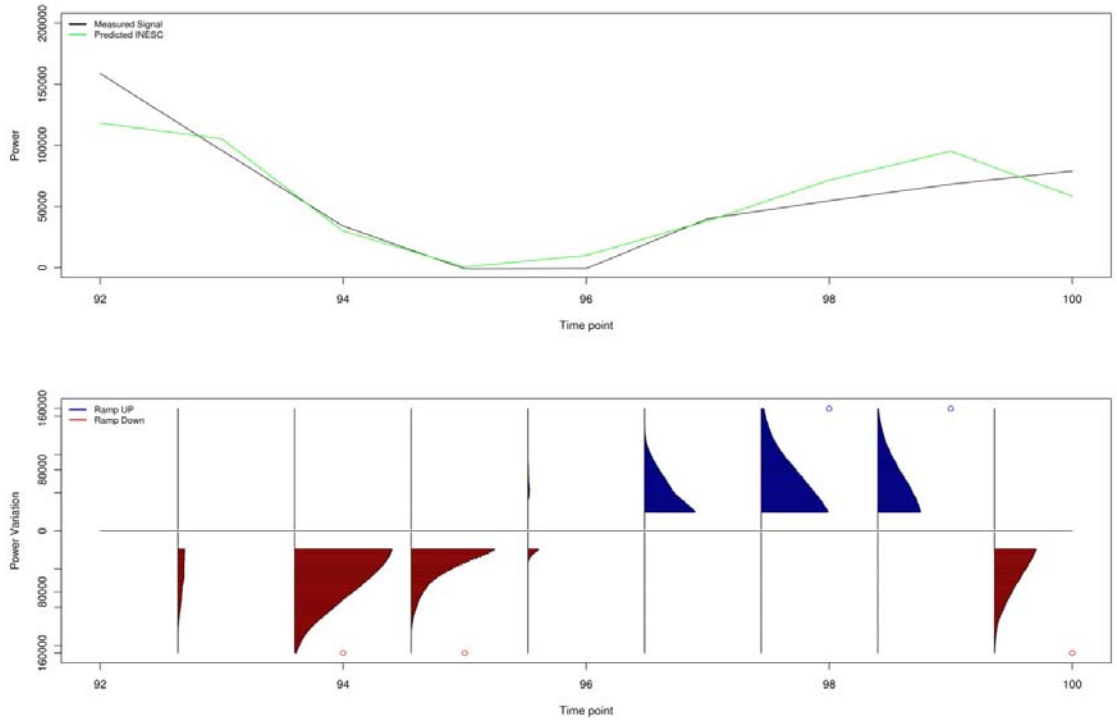

Fig. S9 Histograms of ramp events for one day using a 3h aggregation. Up: wind power point forecast and actual wind farm production. Down: modeling results of Ramp-Up and Ramp-Down events using the new proposed method.

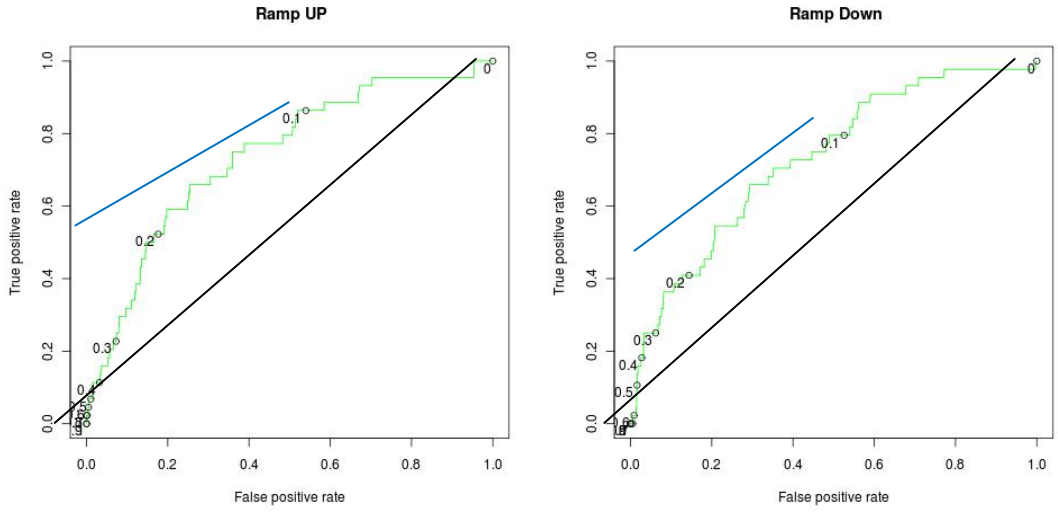

Fig. S10 ROC curves for the new proposed method. Left: ramp-up event. Right: ramp-down event. The blue line is the tangent at the optimum point, according to the configuration used: a false negative alarm cost equal to 200 and a false positive alarm cost of 10.

The main contributions of our ramp forecasting study include the following:

- The new method is independent of a particular ramp event definition and can be implemented using any definition.
- By using a technique to correct phase errors, the proposed method obtained important gains in the forecasting performance when compared to a reference model (i.e., a point forecast).

In summary, this report documents our contributions toward improved statistical methods for WPF. The main results for wind power point, uncertainty, and ramp event forecasting have been illustrated above. Most of the WPF prototypes and algorithms we developed that generated the results presented in this project have been integrated into a research software platform named "ARGUS-PRIMA." More information about the platform can be obtained from Argonne National Laboratory.

This page intentionally blank

1 INTRODUCTION

Wind power forecasting (WPF) provides important inputs to power system operators and electricity market participants. It is therefore not surprising that WPF has attracted increasing interest within the electric power industry. In this report, we document our research on improving statistical WPF algorithms for point, uncertainty, and ramp forecasting. Below, we provide a brief introduction to the research presented in the following chapters. For a detailed overview of the state-of-the-art in wind power forecasting, we refer to [1]. Our related work on the application of WPF in operational decisions is documented in [2].

Point forecasts of wind power are highly dependent on the training criteria used in the statistical algorithms that are used to convert weather forecasts and observational data to a power forecast. In Chapter 2, we explore the application of information theoretic learning (ITL) as opposed to the classical minimum square error (MSE) criterion for point forecasting. In contrast to the MSE criterion, ITL criteria do not assume a Gaussian distribution of the forecasting errors. We investigate to what extent ITL criteria yield better results. In addition, we analyze time-adaptive training algorithms and how they enable WPF algorithms to cope with non-stationary data and, thus, to adapt to new situations without requiring additional offline training of the model. We test the new point forecasting algorithms on two wind farms located in the U.S. Midwest.

Although there have been advancements in deterministic WPF, a single-valued forecast cannot provide information on the dispersion of observations around the predicted value. We argue that it is essential to generate, together with (or as an alternative to) point forecasts, a representation of the wind power uncertainty. Wind power uncertainty representation can take the form of probabilistic forecasts (e.g., probability density function, quantiles), risk indices (e.g., prediction risk index) or scenarios (with spatial and/or temporal dependence). Statistical approaches to uncertainty forecasting basically consist of estimating the uncertainty based on observed forecasting errors. Quantile regression (QR) is currently a commonly used approach in uncertainty forecasting. In Chapter 3, we propose new statistical approaches to the uncertainty estimation problem by employing kernel density forecast (KDF) methods. We use two estimators in both offline and time-adaptive modes, namely, the Nadaraya-Watson (NW) and Quantile-copula (QC) estimators. We conduct detailed tests of the new approaches using QR as a benchmark.

One of the major issues in wind power generation are sudden and large changes of wind power output over a short period of time, namely ramping events. In Chapter 4, we perform a comparative study of existing definitions and methodologies for ramp forecasting. We also introduce a new probabilistic method for ramp event detection. The method starts with a stochastic algorithm that generates wind power scenarios, which are passed through a high-pass filter for ramp detection and estimation of the likelihood of ramp events to happen.

The report is organized as follows: Chapter 2 presents the results of the application of ITL training criteria to deterministic WPF; Chapter 3 reports the study on probabilistic WPF, including new contributions to wind power uncertainty forecasting; Chapter 4 presents a new method to predict and visualize ramp events, comparing it with state-of-the-art methodologies; Chapter 5 briefly summarizes the main findings and contributions of this report.

This page intentionally blank

2 TESTING OF ITL CRITERIA FOR WIND POWER POINT FORECASTS

This chapter covers the following content. After an introductory section, where the point forecasting architecture is presented, in Section 2.2 we describe the data treatment and its preparation — it is detailed general data treatment, which is the same for all of the forecasting metrics used. Section 2.3 describes the process of mapper training, as well as the implementation details of the custom C++ neural network (NN) library, specifically developed for the purpose of this project. This chapter focuses on the developed point forecasting methodology and results; thus, most of this chapter is directly related to the algorithms implemented in this library.

In Section 2.4, the metrics we used are detailed. Section 2.5 defines the performance evaluation metrics (e.g., normalized mean absolute error [NMAE], normalized bias [NBIAS]). The remainder of the chapter presents the results we obtained.

2.1 Introduction

The platform of the algorithms used in this project to analyze wind power point forecast is depicted in Fig. 2-1.

The forecasting process can be functionally separated into three main blocks:

- Retrieval and preparation of input data;
- Training and production of forecasts; and
- Storage and evaluation of results.

Because the forecasting processes are heavily data dependent, proper treatment of the data is crucial. For this reason, the software in this project is supported by a relational database where input data is firstly stored and manipulated. The same database is used to preprocess and prepare the data for the training process. The database also figures as a storage medium for prediction results, allowing comparisons and parameter sensitivity analyses.

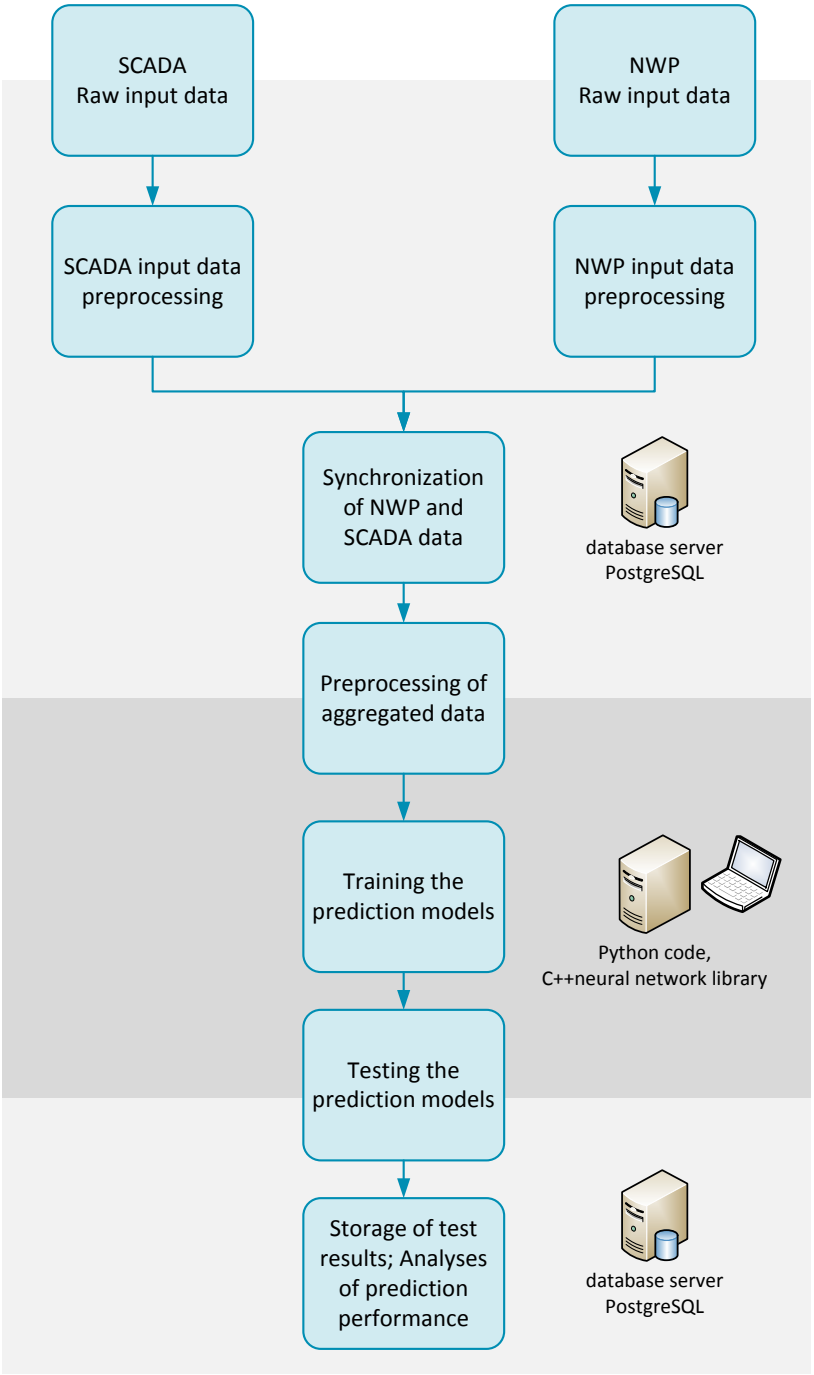

Fig. 2-1 Forecasting process functional architecture.

2.2 General Information – Data Treatment

In order to handle large amounts of input/output data efficiently, a relational database is used for their storage and manipulation. For this project, the PostgreSQL [3] relational database system was chosen. Input data from SCADA and NWP are imported from the input files into the corresponding tables. The integrity of the data is ensured by enforcing relational constraints. After having the data imported, initial preprocessing and aggregation are also performed in the

database, using stored procedures written in the PostgreSQL native PL/pgSQL programming language. The outputs of preprocessing are tables and views that serve as actual inputs to the prediction training system, as shown below (Fig. 2-2).

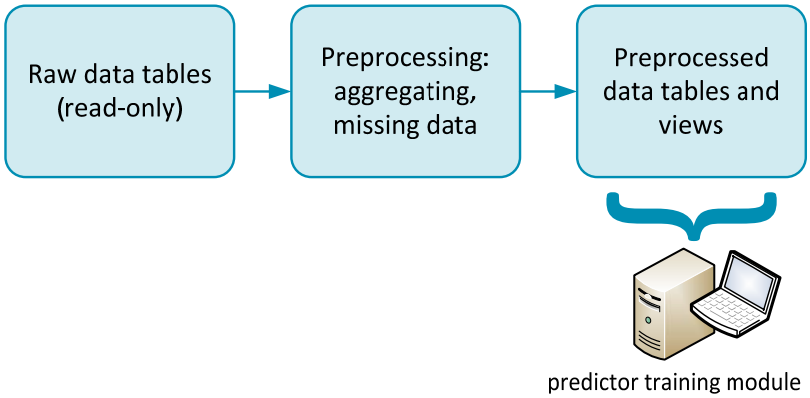

Fig. 2-2 Functional architecture of in-database preprocessing of raw data.

In this project, a W2P — wind-to-power — prediction model is analyzed. The input data required to train such W2P models are:

- Realized wind power, or the prediction targets originated from the wind park's supervisory control and data acquisition (SCADA) system and provided by the wind park operator;
- Numerical weather predictions (NWP) results obtained from Argonne National Laboratory (Argonne). Argonne used the weather research forecast (WRF) model [4] to generate the NWP results with a spatial resolution of 5 km by 5 km over the wind farm area.

The data above have a 10-minute resolution. In order to be used in training, these data are aggregated on a temporal basis so that finer temporal resolution (10-minute) is converted to one that is coarser (hourly). The same aggregation scheme was applied to both SCADA and NWP values, as follows:

$$P_{hour} = \frac{1}{N}\sum_{i=1}^{N} P_{10\,min} \qquad (2\text{-}1)$$

A simple averaging scheme of N measurements during the observed hour is used. Note also that if only complete SCADA measurements are available for a given hour, the number of measurements is N=6 (i.e., in case of missing values, N will be lower).

Besides the temporal aggregation, data aggregation can be performed on a spatial level where a "virtual" wind park is created, corresponding to a geographically related subset of wind turbines. For the purposes of this report, two spatial aggregations have been performed, splitting the data from the large-scale wind park into two wind farms (Wind Park A [WPA] and Wind Park B [WPB]).

Furthermore, in order to train and evaluate wind power forecasts, the complete available data need to be split into two datasets:

- A **training** dataset with targets (desired values) known in advance — this dataset will be used exclusively for training; and
- A **testing** dataset where the realizations are predicted by the trained W2P — this way, the testing mimics the actual application of the forecasting process.

Note that if online learning is used, the W2P model continues to learn during the testing phase and is constantly correcting its internal weights. Online learning is one way of dealing with the non-stationary characteristics of the wind.

The complete dataset (SCADA and NWPs) available for this project corresponds to the period between January 1, 2009, and February 20, 2010. Hence, the following data partition was used:

- **January 1, 2009 – June 1, 2009 — training dataset**, with 4,992 total hourly samples;
- **June 1, 2009 – February 20, 2010 — testing dataset**, with 4,680 total hourly samples.

In the input NWP data, there are 11 NWP points geographically distributed over the wind park area. A single NWP reference point was chosen for each of the wind parks. For WPA, the data from NWP point **8** were used. For WPB, NWP point **6** was chosen as the reference point. These two NWP points are both located within the wind farm.

With regard to forecasting horizon, the complete NWP data consists of two 48-hour forecasts per day. For the purposes of this chapter, only the morning NWP forecasts, launched at 6:00 AM, were used. The temporal horizon used in forecasts is "day-ahead," so the predictions are developed for the following day. This means that the forecasts were created for the temporal horizon of t+18 up to t+42 hours, and there is a single forecast available for each time step, launched at 6:00 AM on the previous day.

2.3 W2P Predictor Training

2.3.1 General Information on W2P Training

The predictor based on a neural network is an example of a Wind to Power or W2P model. For a detailed description of various prediction models, see [1].

In practice, a W2P model takes the numerical weather (NWP) forecasts as inputs and gives the predicted wind power as output. The output **y** of a W2P **g** is a function of inputs **x** and W2P internal weights **w**, such that:

$$g(x, w) = y \tag{2-2}$$

In the case of wind power prediction, the inputs correspond to the vector of explanatory variables, such as the NWPs of wind speed and direction. The output is generally the wind power prediction for a certain horizon.

Training of such W2P corresponds to searching for a set of weights that performs the best mapping, according to the performance measure related to the error $e = T - y$. The generalized process of W2P training is depicted in Fig. 2-3.

In other words, a W2P is subjected to the process of supervised training, where its weights, w, are adjusted by a training algorithm in order to produce a known output, y, from a known input, x. The set of known inputs and corresponding known outputs is a training set. The performance criterion is the metric used to evaluate the predictor's performance.

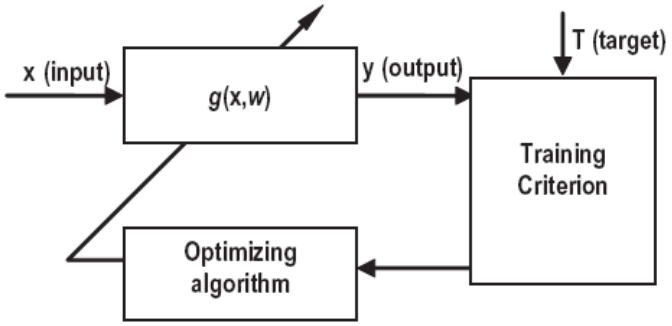

Fig. 2-3 A generalized representation of W2P training.

A neural network consists of several layers of neurons. The neurons are linked by synapses, passing the signal from one neuron to another and conditioning it by synapse weight. A neuron's output is the effect of an activation function acting on a linear combination of inputs from the previous layer and conditioned by synapses linking the neurons. A comprehensive introduction to neural networks is given in [5]. The process of training a neural network consists of making adjustments to the synapse weights in order to achieve better performance, according to the training criterion. It is worth noting that the neuron x_o in Fig. 2-4 always emits 1 — this is the bias neuron, and the synapse weight w_{k0} corresponds to the value of bias for this neuron.

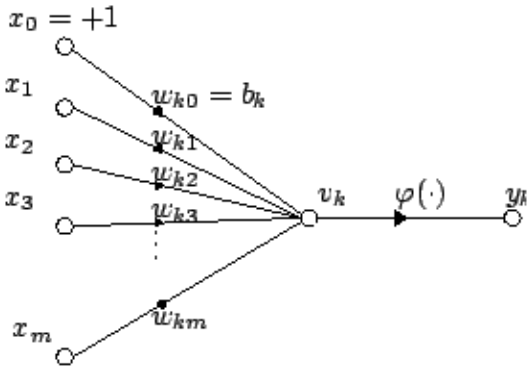

Fig. 2-4 Neuron representation in a neural network (Source: Wikipedia)

The neural network *backpropagation algorithm* applies a correction to synapse weights that is proportional to the partial derivative of the error with regard to the weight. In other words, it is a gradient descent algorithm in the space of weights: it seeks a direction for weight change that reduces the value of error (Fig. 2-5).

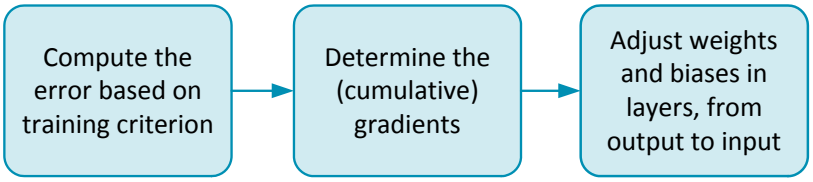

Fig. 2-5 Illustration of backpropagation algorithm process.

Furthermore, neural network training algorithms in general can be divided in two main classes:

- **Batch** training algorithms (require a set of samples), and
- **Incremental** training algorithms (used in sample-by-sample training).

The main difference between batch and incremental algorithms lies in the manner of presenting the training samples and weight and bias adjustments. While batch algorithms perform adjustments to network weights and biases only after the *whole batch* is presented to the network, the incremental training algorithms perform such adjustments after presenting *each* sample. In other words, for batch algorithms the gradients are cumulative and adjustments are performed after all of the samples have been presented. The batch set is typically the whole training dataset.

A simple backpropagation algorithm directly uses the correction based on (cumulative, in the case of batch methods) gradient:

$$\Delta w_{t+1} = -\eta \frac{\partial e_t}{\partial w_t} \tag{2-3}$$

where the change from epoch t to epoch $t+1$ is directly proportional to the gradient and conditioned by the learning rate η. A small learning rate might lead the network to a local optimum and deliver unsatisfactory performance, while a higher learning rate means the change in weights will be more intensive, which may lead to unstable oscillatory behavior in the learning process, and thus a failure of convergence of W2P training.

A typical measure for avoiding such problems is to add a *momentum* term μ to the weight update equation:

$$\Delta w_{t+1} = -\eta \frac{\partial e_t}{\partial w_t} + \mu(w_t - w_{t-1}) \tag{2-4}$$

The momentum term works as a constant in a feedback loop around Δw. If the error gradient keeps its sign over iterations, the momentum term will increase the steady convergence. On the other hand, if the sign of the gradient changes between consecutive iterations, the momentum term will attenuate the change, stabilizing the convergence process and avoiding oscillation. For the process to converge, it is obvious that the momentum has to be less than one, $|\mu| \leq 1$.

Besides the classic backpropagation algorithm, there are other rules that have been devised to handle the weight update process. Among the batch training algorithms, the iRPROP [6] algorithm is considered to be a very robust and efficient algorithm for the mean square error training criterion. Typically, it is notably more efficient per training iteration than classic

backpropagation. For this reason, the iRPROP algorithm with the classic MSE is used as a base for comparisons between algorithm performances. While it is based on the same fundamental rules, iRPROP uses a different weight update rule, so it does not use the learning and momentum parameters in the classic form.

Even though batch training algorithms have, in general, more favorable characteristics if compared with those that are incremental, the latter are needed for adaptive training. Adaptive methods of training are necessary for wind power forecasting. Any W2P model trained offline will display, after some time and after the imminent appearance of so-called concept drift, a pattern of growing error in prediction values. To deal with such non-stationary behavior, adaptive methods are needed. For this reason, the methodology implemented in the neural network library implements incremental training algorithms in addition to the batch algorithms. In addition, the incremental algorithms implement the randomization of the learning patterns when incremental training methods are used for a set of patterns. Because the incremental training algorithms present the samples one by one, presenting the samples from a batch in the same order would lead to "overfitting" the network for some samples. This outcome is avoided by shuffling the order of samples.

Considering the software implementation, the supervised training is implemented in a combination of the Python programming language and C++ programming language. The Python language enables simple interaction with the relational database in order to retrieve and store the relevant data, and the C++ neural network library is responsible for computationally more intensive tasks. The evaluation of training criteria is the most demanding task in the W2P training; thus, most of it is implemented in C++. Data management is somewhat split between Python code and the PostgreSQL database in order to maintain the flexibility of the training process. The code is organized in a manner that enables predictors to use an arbitrary number of explanatory parameters, and various predictors can be trained and used in parallel. Each of these could rely on its own set of input parameters (explanatory variables).

2.3.1.1 Neural Network Library Implementaion Details

As stated before, the neural network is implemented as a C++ library, and several classes are "exposed" to Python. The Python code has control over the neural network training process, while the computationally demanding tasks are implemented in C++. The library represents a software implementation of network structures and algorithms, and it was specifically developed for the purpose of this project.

There are four main classes in the C++ library. The code structure of implemented C++ classes resembles the actual structure of the neural network, as follows:

- The **Neuron** class implements the functions related to the neuron (i.e., bias adjustment), references the associated synapses, and implements calculation of various error metrics;
- The **Synapse** class is primarily responsible for storing the synapse weight (a synapse is a connection between two neurons) and its adjustments;
- The **NeuralNetwork** class encapsulates the entire network and implementation of the training algorithms; and
- The **Pattern** class represents a training pattern (consisting of a set of inputs x and desired outputs T).

By means of the Boost.Python [7] interface, the neural network library interfaces with the rest of the forecasting method development platform. The NeuralNetwork class is "visible" from the Python code that prepares the data, and the dataset is, after extraction from the database, prepared as a vector of Patterns. The library also implements a "helper" **RandomGen** class that implements high-quality random number generators whose routines are based on the Boost.Random library.

There is a significant difference in current practice as compared to the neural network architecture first developed for this project. The library used to deliver the results for this chapter is a newly developed library without dependencies on any pre-built neural network libraries, so the current version of the NN library is developed specifically for this project and customized for the purpose of specific W2P training.

2.3.2 Training Error Measures

For the supervised W2P training, a criterion related with forecast error is required. The error is defined as the difference between the desired value (target) and the W2P model output:

$$e = T - y \tag{2-5}$$

In this project, the in-training performance evaluation criteria introduce Information Theoretic Learning (ITL) error measures [8]. The ITL measures include Gaussian kernels, and in the subsequent formulae, a Gaussian kernel is referenced as $k_\sigma(x) = \frac{1}{2\pi\sigma} e^{-\frac{x}{2\sigma}}$.

2.3.2.1 MSE – Minimum Square Error

This is the classical neural network training criterion that minimizes the variance of the error distribution and has the form:

$$MSE(e) \Leftrightarrow \min \frac{1}{\mathrm{N}} \sum_{i=1}^{N} e_i^2 \tag{2-6}$$

where $e_i = (T_i - y_i)$ is the error of sample i relative to target value T_i (and output y_i), and N is the total number of training samples. The minimum square error criterion is based on the assumption that the errors form a Gaussian distribution.

2.3.2.2 MEE – Minimum Error Entropy

This criterion minimizes the entropy of the errors, e, which is equivalent to maximizing the information potential, V, of the dataset:

$$MEE(e) \Leftrightarrow \min[-\log \mathrm{V(y)}] \quad \Leftrightarrow \max \frac{1}{\mathrm{N}^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathrm{k}_{2\sigma^2}(e_i - e_j) \tag{2-7}$$

where $\mathrm{V(y)} = \frac{1}{\mathrm{N}^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathrm{k}_{2\sigma^2}(e_i - e_j)$ and $k_{2\sigma^2}$ is the Gaussian kernel with bandwidth $2\sigma^2$.

The general idea behind minimizing error entropy is to compress the error distribution into a Dirac function, and then, by correcting the bias, the errors would be eliminated.

Only where errors really are Gaussian, the Gaussian distribution is a good approximation of error distribution, so the MSE criterion yields good performance. On the other hand, the MEE criterion does not impose Gaussian assumptions on the gaussianity of the error distribution; thus, it should perform well even for non-Gaussian error distributions.

The primary disadvantage of MEE criterion is the computational burden it introduces in calculation. The double sum in the formulation of error function requires, for each training sample prediction error, calculation of Gaussian kernel values for error differences of that sample with all of the other samples' errors. This requirement means that the algorithmic complexity of such a process is practically $O(n^2)$, which makes the problem intractable for use with a large number of samples, even though the neural network training library may use parallel (multithreaded) computation to calculate the estimation of error distributions.

An adequate approximation is the so-called batch-sequential [9] training algorithm that randomly divides the training dataset into several smaller subsets, then calculates the error entropy for each of the subsets, instead of doing so for the whole training set. After calculation of the error entropy of the subsets, the network weights are updated.

The random partition of the dataset is initialized at the beginning of each training epoch, that is, the partition is not kept constant over the epochs. This process requires an additional parameter—subset size. For the presented examples, subset size is set to approximately one-third of the whole training dataset (i.e., 1,500 patterns).

The reasoning behind such an algorithm is combining the batch mode where an update of the weights is made only once after presenting all of the samples and calculating the cumulative gradient, with an incremental mode of training where weights are updated after presentation of each sample. The lowered size of subsets in comparison with the whole training set results in much faster computation of error entropy and consequently means that the error entropy calculation is tractable for larger datasets.

2.3.2.3 MCC – Maximum Correntropy Criteria

This criterion is based on *correntropy* measure and may be given by:

$$MCC(e) \Leftrightarrow \max \frac{1}{\mathrm{N}} \sum_{i=1}^{N} \mathrm{k}_{\sigma}(e_i) \tag{2-8}$$

Correntropy is a generalized similarity measure between two arbitrary scalar random variables. It is directly related to the probability of how similar two random variables are in a neighborhood of the joint space defined by the kernel bandwidth. If a Gaussian kernel is used, then the correntropy measure is equivalent to Euclidean norm, when the error is close to zero. If the error increases, it first becomes similar to a L1 norm and, for very large errors, correntropy becomes insensitive. This means that correntropy is a robust measure. However, the desired kernel size is important: for very large kernel sizes, correntropy behaves analogously to the MSE metric. Typically, in the beginning of the training with MCC, it should be started with large kernel sizes so that the network would not ignore the presented patterns. This approach is also followed in the subsequent results – Gaussian kernel size is reduced after the initial 25% of training epochs.

2.3.2.4 MEEF – Minimum Error Entropy with Fiducial Points

A problem of the minimum error entropy criterion is that it is not restricted to a zero mean. This criterion adds a MCC term to the MEE training criterion, in order to deal with the lack of constraint of the mean value error in the latter, as:

$$MEEF(e) \Leftrightarrow \max \left(\gamma \frac{1}{\mathrm{N}} \sum_{i=1}^{N} \mathrm{k}_{\sigma^2}(e_i) + (1-\gamma) \frac{1}{\mathrm{N}^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathrm{k}_{2\sigma^2}(e_j - e_i) \right) \quad (2\text{-}9)$$

where γ is a weighting constant between 0 and 1. This training criterion aims to anchor the error distribution to a zero mean by defining a compromise between minimizing entropy and maximizing correntropy through a cost function.

2.3.2.5 cMCC– Centered Maximum Correntropy Criterion

This criterion is estimated as the maximum difference between MCC and MEE:

$$CMCC(e) \Leftrightarrow \max \left(\frac{1}{\mathrm{N}} \sum_{i=1}^{N} \mathrm{k}_{\sigma}(e_i) - \frac{1}{\mathrm{N}^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathrm{k}_{2\sigma^2}(e_j - e_i) \right) \quad (2\text{-}10)$$

Further information on the underlying criteria can be found in [10] and [11]. However, the cMCC criterion as a difference between the MCC and MEE criteria is introduced in the scope of this project — this criteria aims to exploit the benefits of the MCC and MEE criteria, avoiding the worsening of bias when MEE is used while also keeping the robustness MEE criterion offers.

2.3.2.6 Incremental Training and Error Entropy Criteria

For criteria relying on error entropy, multiple samples are inherently required. A construction of an incremental training method requires a recursive setup for calculation of information.

The formulation of the recursive update of information potential using a forgetting factor is:

$$IP_{t+1} = (1-\lambda) IP_t + \frac{\lambda}{M} \sum_{i=k-M+1}^{k} k_{\sigma}(e_i - e_{k+1}) \quad (2\text{-}11)$$

where λ is a forgetting factor and M is the size of the window for the recursive update. When a new sample is obtained for the epoch t+1, it conditions the existing value of information potential using the above formulation.

2.3.3 Discrete Kalman Filters in WPF

The experiences from the discipline of statistics show that a combination of multiple *diverse* forecasting models gives favorable results — capable of surpassing the performance of each of the models separately. The main reasoning behind this finding is that the nature of the errors of different models is also different; thus, if an appropriate aggregation scheme for multiple models is used, a combination may surpass the performance of any single model. The condition for combining models is that individual models should have a substantial level of disagreement.

With regard to this project, several neural networks with different cost functions can be seen as different models. Their forecasts are then combined by using a Kalman filtering method.

A Kalman filter is an optimal recursive data processing algorithm that uses noisy measurements (with random variations) and other inaccuracies in order to obtain results that tend to be closer to the true values of the measurements. In the case of WPF, a Kalman filter would try to prefer the "better" model for a given condition.

The Kalman filter produces estimates of the true values of measurements and their calculated values by predicting a value and estimating its uncertainty and weighted average, as well as computing the measured value. The highest weight is given to the value with the least amount of uncertainty. The estimates produced by this method tend to be closer to the true values than they are to the original measurements, because the weighted average has lower estimated uncertainty than either of the values that went into the weighted average.

The Kalman filter has two distinct phases: predict and update. The former is a time update in which the estimate of the state from the previous time step, $k-1$, is used to predict the state at the current timestep, k. This *a priori* state estimate, $\hat{x}^-$, does not include observation information from the current timestep. Therefore, $\hat{x}^-$ and the *a priori* error covariance, P^-, are respectively determined according to:

$$\hat{x}_k^- = A\,\hat{x}_{k-1} + B\,u_k \tag{2-12}$$

$$P_k^- = A\,P_{k-1}A^T + Q \tag{2-13}$$

where u is the input vector, A is the transport matrix, B the input matrix, and Q is the process variance.

In the update phase, on the other hand, the current *a priori* prediction is combined with current observation information to refine the state estimate. This improved prediction is termed the *a posteriori* state estimate. Hence, the blending factor (or Kalman gain), K, the *a posteriori* state estimate, $\hat{x}$, and the *a posteriori* error, P, are respectively given by:

$$K_k = P_k^-\,H^T\,(H\,P_k^-\,H^T + R)^{-1} \tag{2-14}$$

$$\hat{x}_k = \hat{x}_k^- + K_k\,(z_k - Hx_k^-) \tag{2-15}$$

$$P_k = (I - K_kH)\,P_k^- \tag{2-16}$$

where z are the noisy measurements, H is a matrix, I is the identity matrix, and R an estimate of the measurement variance.

Further information about Kalman filters can be found in [12] and [13]. While the algorithmic basis for the use of Kalman filters is developed in the project, the preliminary results are subject to revision at the moment. Additional data might be necessary for this purpose.

2.3.4 Prediction Performance Evaluation Metrics

In order to systematically evaluate forecasting performance, one has to establish a set of metrics. The following error metrics conform to a commonly accepted definition of systematic evaluation

metrics for prediction performance evaluation.

The *normalized prediction error* is defined as

$$\epsilon(t+k|t) = \frac{1}{P_{inst}}\left(P(t+k) - \hat{P}(t+k|t)\right) \quad (2\text{-}17)$$

which is the difference between target (realized value at time t+k, $P(t+k)$) and forecasted value $\hat{P}(t+k|t)$, divided by the wind park installed power, P_{inst}. One can subsequently define the following metrics used to evaluate the quality of the forecasts, as follows:

NMAE – Normalized Mean Absolute Error

$$NMAE = \frac{1}{N}\sum_{t=1}^{N}|\epsilon(t+k|t)| \quad (2\text{-}18)$$

NRMSE – Normalized Root Mean Square Error

$$NRMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(\epsilon(t+k|t)\right)^2} \quad (2\text{-}19)$$

An alternative use of root mean square error (RMSE) is to consider the Standard Deviation of the Errors (SDE), or its normalized value:

$$NSDE_k = \sqrt{\frac{\sum_{t=1}^{N}\left[\epsilon_{t+k|t} - \bar{\epsilon}_k\right]^2}{N}} \quad (2\text{-}20)$$

Because the SDE criterion is an estimate of the standard deviation of the error distribution, only the random error contributes to the SDE criterion.

NBIAS – Normalized bias (systematic error)

$$NBIAS = \bar{\epsilon} = \frac{1}{N}\sum_{t=1}^{N}\epsilon(t+k|t) \quad (2\text{-}21)$$

All of the above measures depend on the absolute value of the error and thus do not indicate whether the prediction has a systematic error. BIAS measures exactly such kinds of systematic errors.

In the following chapters, the above evaluation metrics are represented for each of the implemented training criteria and for each of the two wind parks. The evaluation period corresponds to the entire testing period. Furthermore, the results present the histogram representation of the forecasting errors to show the shape of error distribution. The comparison of prediction reliability is also indicated through plotting occurrences of forecasted values versus the occurrences of production values for different ranges of installed power.

2.4 Wind Power Point Forecasting Results

In this section, the results of the new methodology for wind power point forecasting are presented. All of the metrics described previously are analyzed. As described in Section 2.2 for all of the cases presented, the data treatment is kept constant; thus, the only difference between the following cases is related to the choice of training algorithm and the evaluation metrics.

The W2P prediction errors inevitably depend on the NWP data. For this reason, the performance of prediction assessment is limited to comparison between different criteria. While the absolute performance measures are visible from the following graphs, it may not be adequate to compare these results directly to forecasts from other systems, using different NWP models and different methods of data preprocessing and postprocessing.

In other words, a direct performance comparison may be misleading given that for a W2P model, one of the crucial points is the quality and treatment of input data. The idea behind the following results was to present the advantage of using ITL criteria in W2P training for a large-scale wind farm in the United States. The data has been split into two "sub-wind" farms, and the following paragraphs illustrate the results obtained for both wind parks.

2.4.1 Wind Farm A

This wind farm is the larger of the two and has about twice as much installed capacity as wind farm B. In all of the cases presented subsequently, 800 iterations (epochs) of the training algorithm were used, and the input data were the same.

2.4.1.1 Minimum Square Error – MSE

For the offline training case, 800 iterations of the iRPROP training algorithm were used. For the iRPROP training algorithm the learning rate parameter has no effect given that it uses an adaptive method of setting the learning rate (Figs. 2-6 through 2-9).

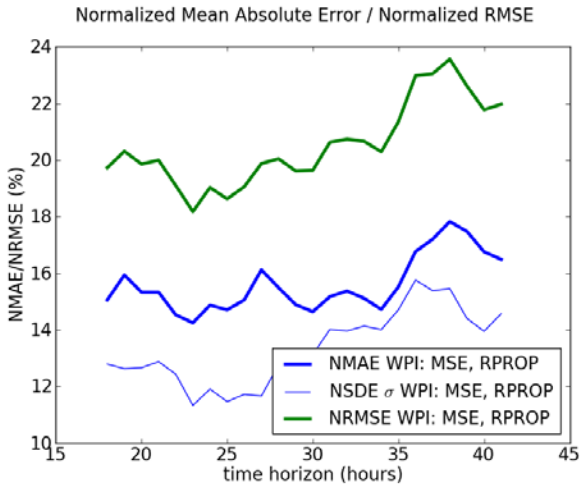

Fig. 2-6 NMAE and NRMSE, for offline training, in Wind Farm A – MSE.

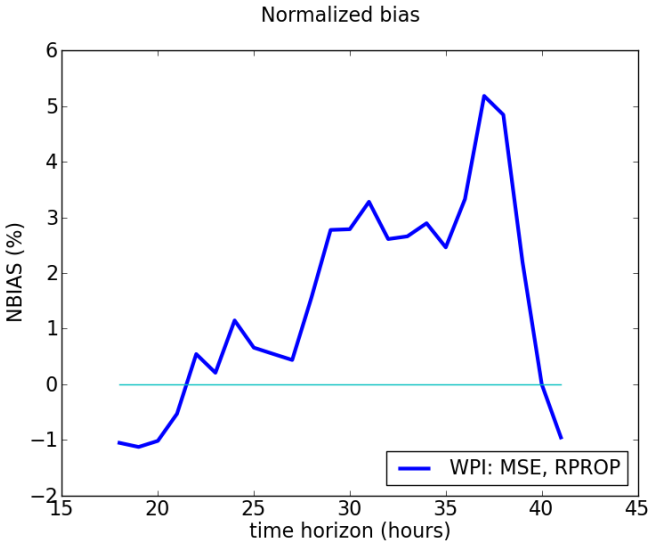

Fig. 2-7 NBIAS for offline training, in Wind Farm A – MSE.

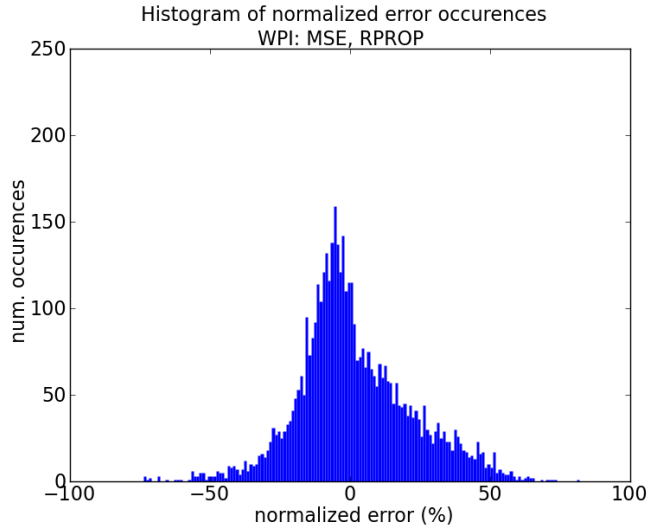

Fig. 2-8 Histogram of error occurrences in Wind Farm A – MSE.

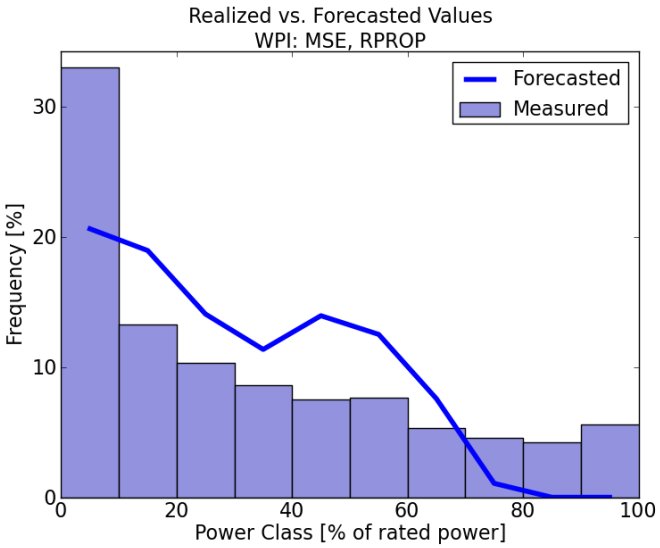

Fig. 2-9 Frequency of occurrence of forecasted and measured values, Wind Farm A – MSE.

From the graphs above, it is evident that the network trained with the MSE criterion is weak in predicting the common power range close to zero, as well as the range in the proximity of nominal power. The network also exhibits the positive total bias of the forecasts. Note that if the error is defined as $e = T - y$, the positive bias means underestimation.

In Figs. 2-10 through 2-12, note that while the introduction of online learning does not have a strong effect on NMAE, it lowers the bias of MSE-based predictions and helps the network approximate the power distribution.

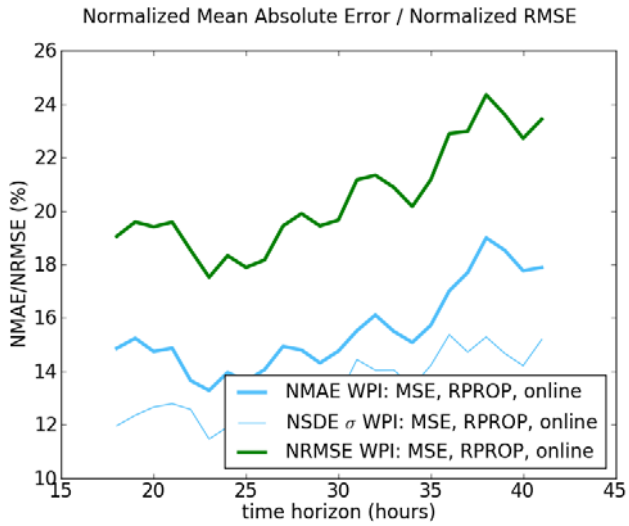

Fig. 2-10 NMAE and NRMSE, for online training, in Wind Farm A – MSE.

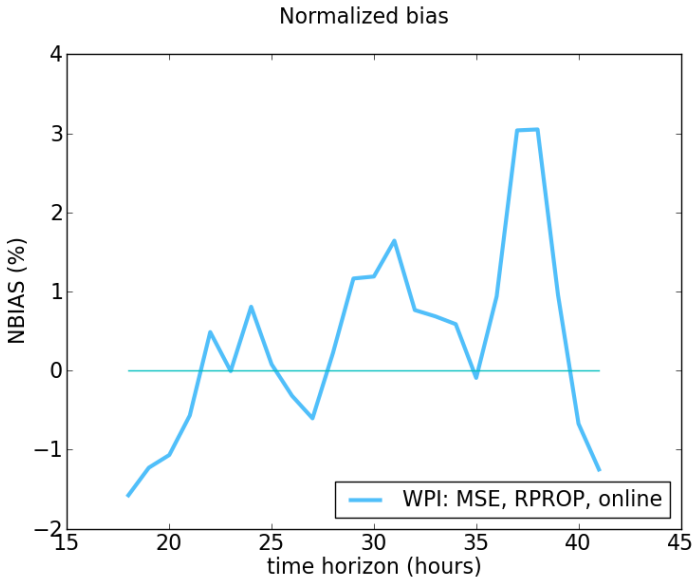

Fig. 2-11 NBIAS for online training, in Wind Farm A – MSE.

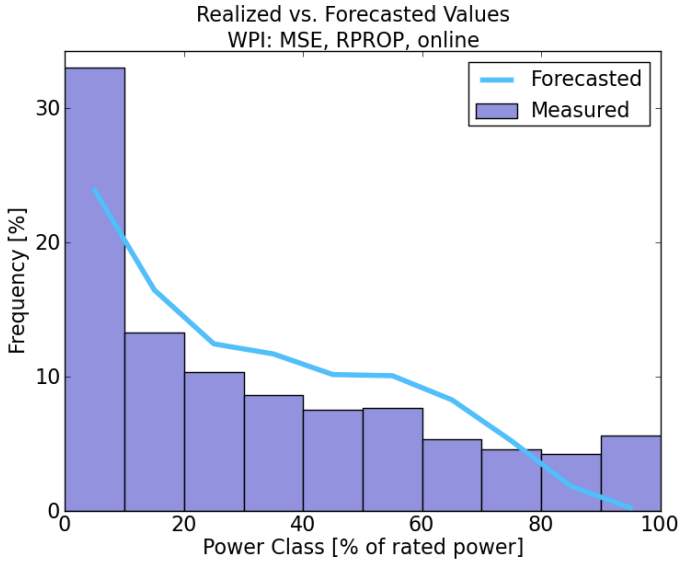

Fig. 2-12 Frequency of occurrence of forecasted and measured values, Wind Farm A – MSE.

2.4.1.2 Maximum Correntropy Criterion – MCC

In the following examples, a Gaussian kernel size σ_{MCC} is lowered after the first 200 iterations, and a classic batch backpropagation training algorithm is used instead of iRPROP. The iRPROP algorithm is considered to be a very efficient training algorithm for MSE criterion. It uses different weight-updating rules and exhibits better per-epoch performance than "pure" batch backpropagation when MSE criterion is used. However, in spite of being more efficient in MSE training, it exhibits lower performance when MCC is used. The classic batch backpropagation algorithm requires setting of the learning rate parameter, which was set to 0.1 (Figs. 2-13 through 2-16).

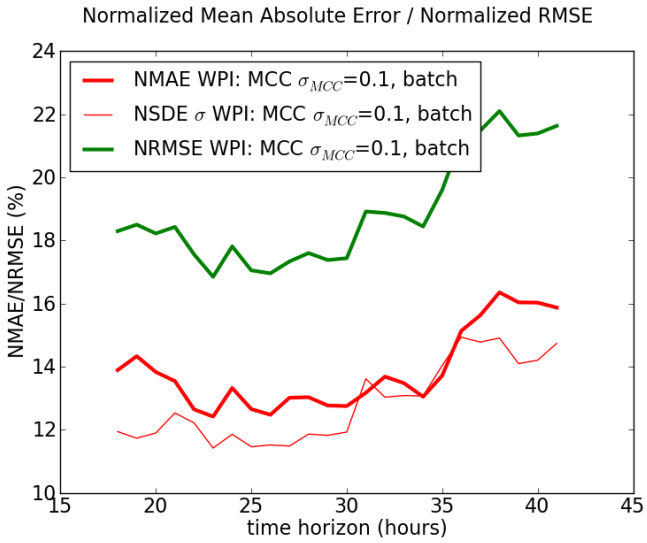

Fig. 2-13 NMAE and NRMSE, for offline training, in Wind Farm A – MCC.

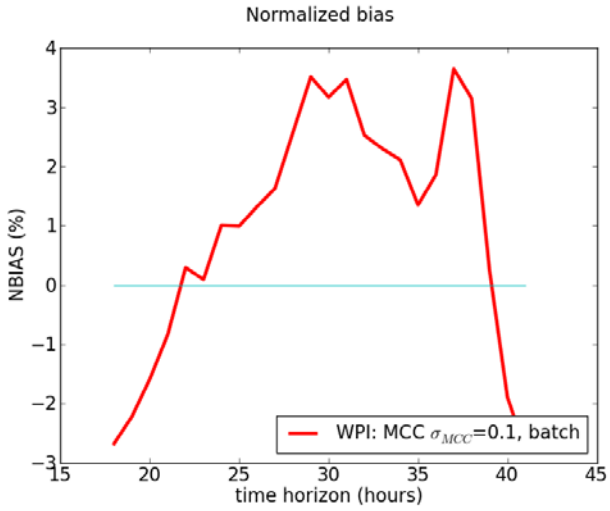

Fig. 2-14 NBIAS for offline training, in Wind Farm A – MCC.

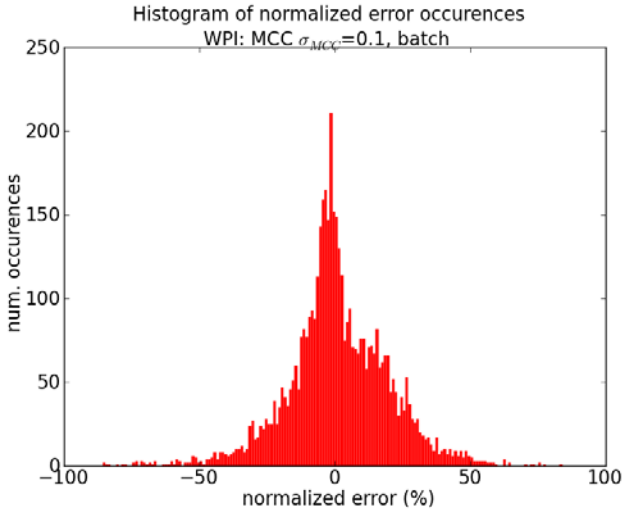

Fig. 2-15 Histogram of error occurrences in Wind Farm A – MCC.

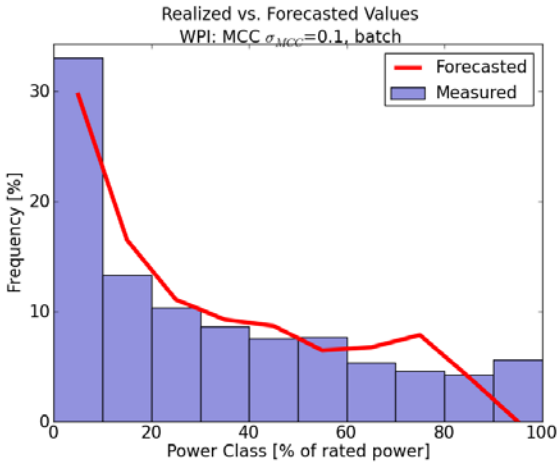

Fig. 2-16 Frequency of occurrence of forecasted and measured values, Wind Farm A – MCC.

If compared with the MSE criterion, the MCC successfully achieves a higher concentration of errors around zero, which is exactly the desired effect and is evident from the lower NMAE error for MCC in comparison with MSE. The effect is even more visible if frequency of occurrence is observed: MCC criterion more closely follows the occurrences of measured values, that is, the distribution of forecasted power is closer to realized values than with MSE criterion.

When the MSE criterion is used, classic batch backpropagation requires more epochs to converge than iRPROP requires. The iRPROP algorithm with MSE performs better than batch backpropagation because of heuristic rules it uses in weight updating. One might conclude that for the MCC criterion, iRPROP should also perform better. However, this is not the case, and it seems that the heuristic rules are not as suitable for MCC criterion. Thus, for MCC the classic batch backpropagation algorithm was used instead (Figs. 2-17 through 2-19).

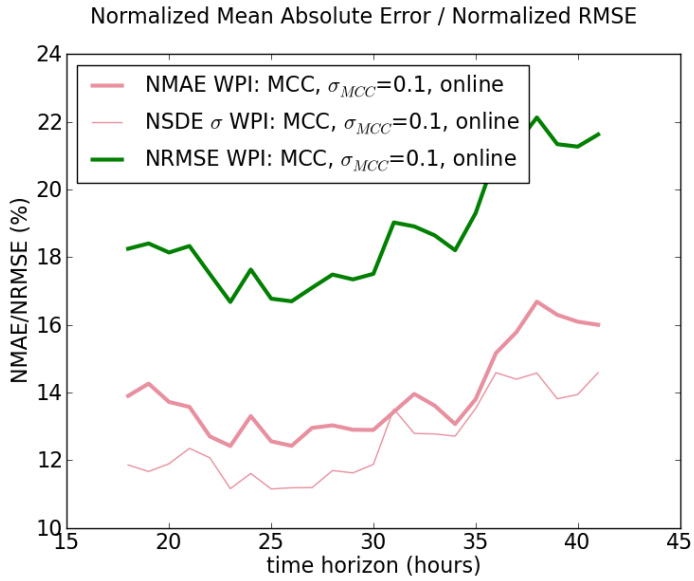

Fig. 2-17 NMAE and NRMSE, for online training, in Wind Farm A – MCC.

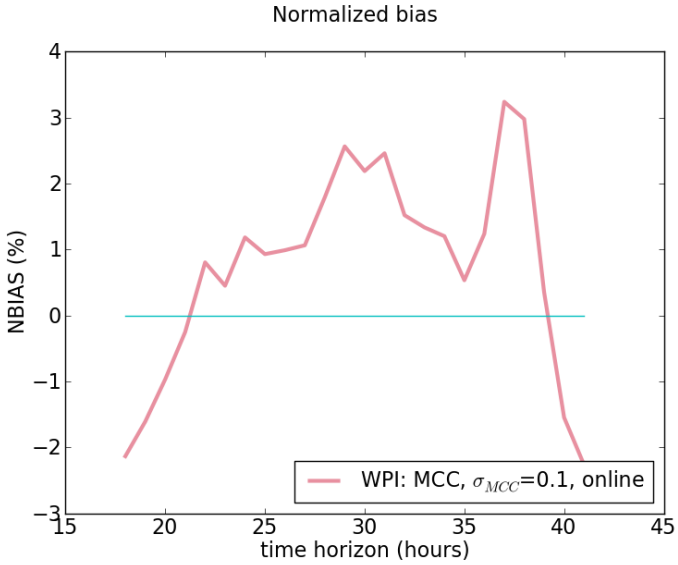

Fig. 2-18 NBIAS for online training, in Wind Farm A – MCC.

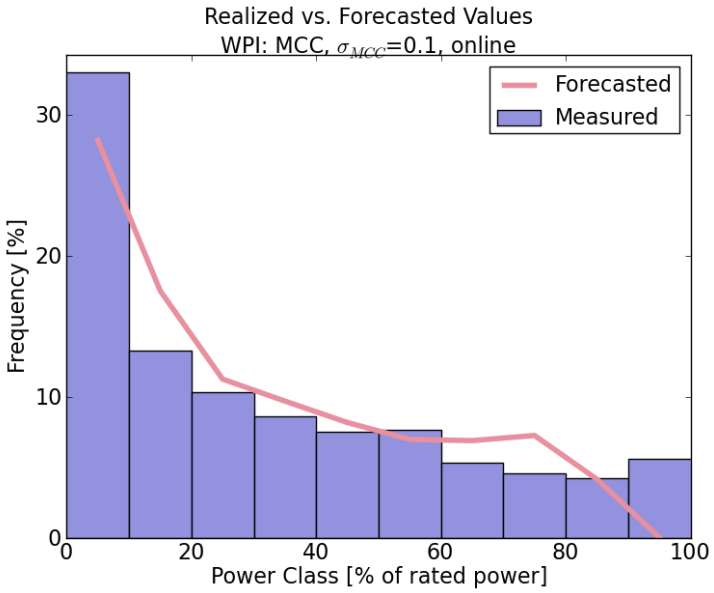

Fig. 2-19 Frequency of occurrence of forecasted and measured values, Wind Farm A – MCC.

Because the offline training succeeds in achieving good performance of the W2P model, the introduction of online training does not increase the performance significantly — only the bias is slightly lower. In addition, the testing dataset is relatively short (6 months only) so the effects of online training are not very visible.

2.4.1.3 Minimum Error Entropy Criterion – MEE

Figs. 2-20 through 2-23 capture findings related to the MEE criterion.

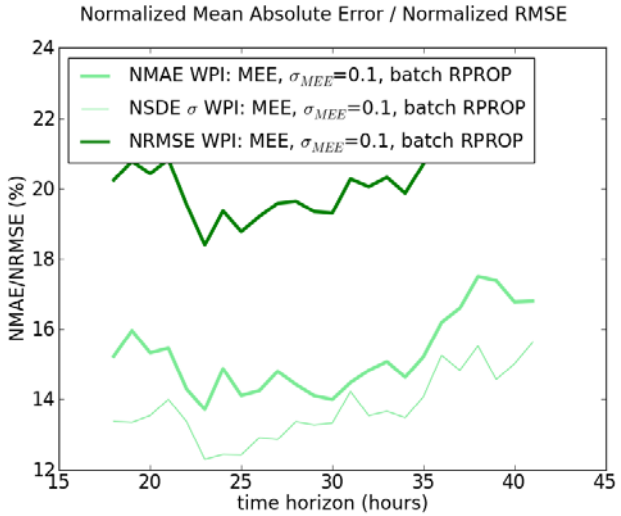

Fig. 2-20 NMAE and NRMSE, offline training, in Wind Farm A – MEE.

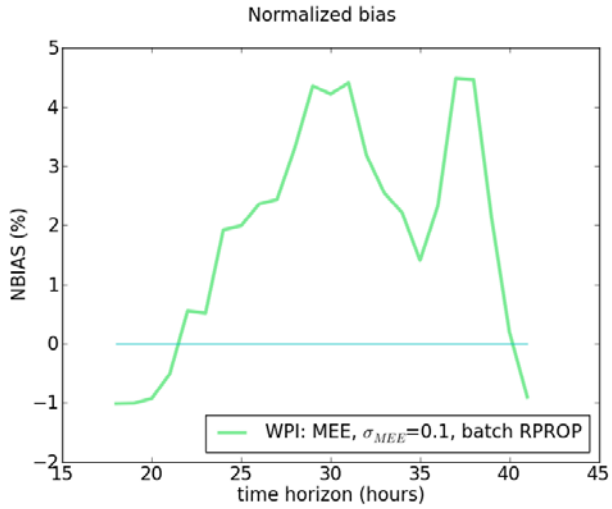

Fig. 2-21 NBIAS for offline training, in Wind Farm A – MEE.

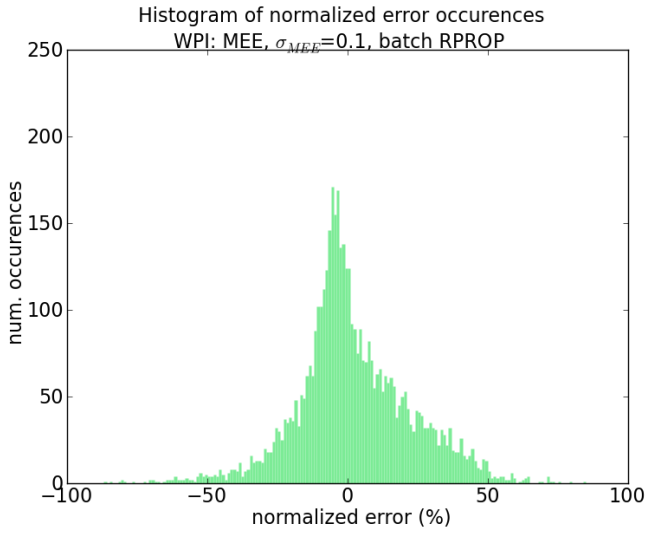

Fig. 2-22 Histogram of error occurrences in Wind Farm A – MEE.

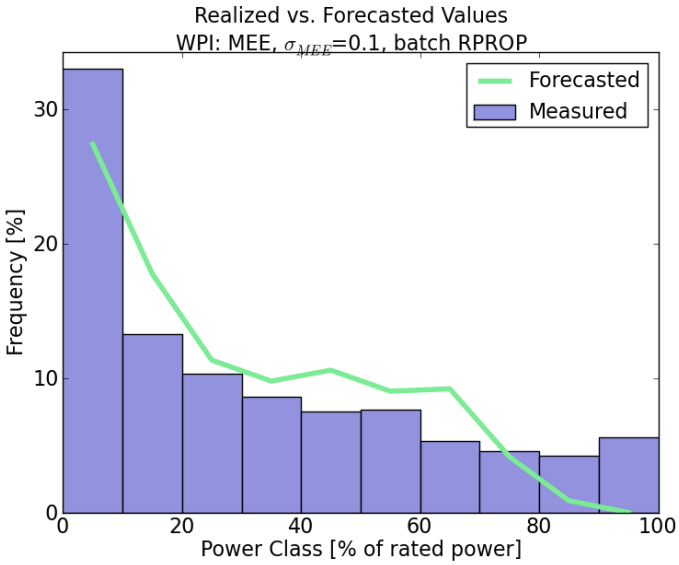

Fig. 2-23 Frequency of occurrence of forecasted and measured values, Wind Farm A – MEE.

It is known from the theory of ITL that the "pure" MEE criterion is insensitive to mean. The result is confirmed in practice given that the MEE criterion, when applied in an offline setting, requires a final step to correct the bias. However, such correction is not as efficient as the corrections to bias made during the training, as employed in MSE and MCC criteria — a finding that is visible from the BIAS results. The MEE criterion is still more successful than the MSE criterion is in following the distribution of occurrence of measured values. In the above example, a classic batch iRPROP was used; thus, the above computation required substantially more computational effort than did the comparable MSE and MCC training.

2.4.1.4 Minimum Error Entropy with Fiducial Points – MEEF

Figs. 2-24 through 2-27 capture findings related to the MEEF criterion.

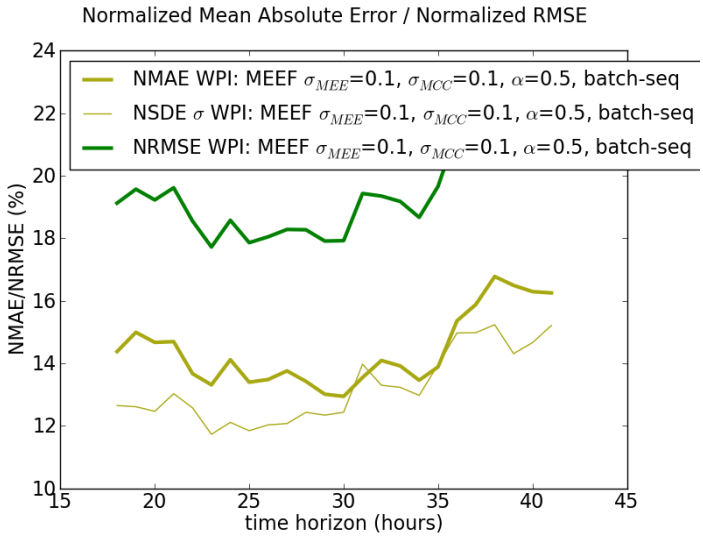

Fig. 2-24 NMAE and NRMSE, offline training, in Wind Farm A – MEEF.

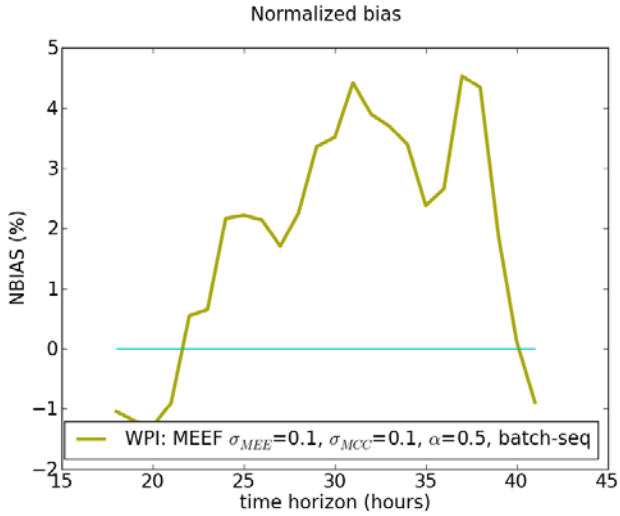

Fig. 2-25 NBIAS for offline training, in Wind Farm A – MEEF.

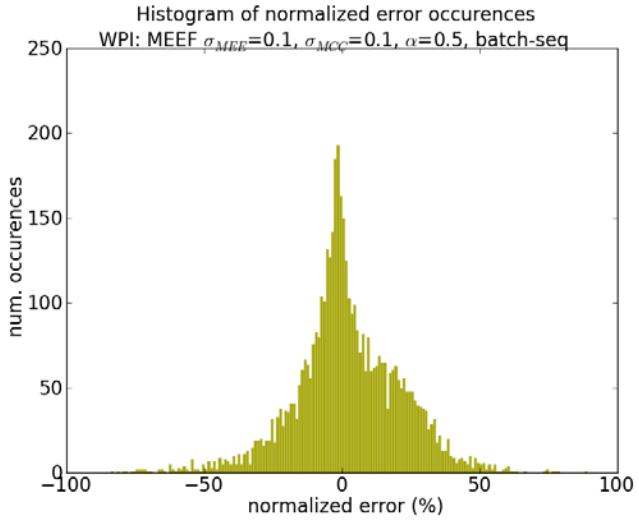

Fig. 2-26 Histogram of error occurrences in Wind Farm A – MEEF.

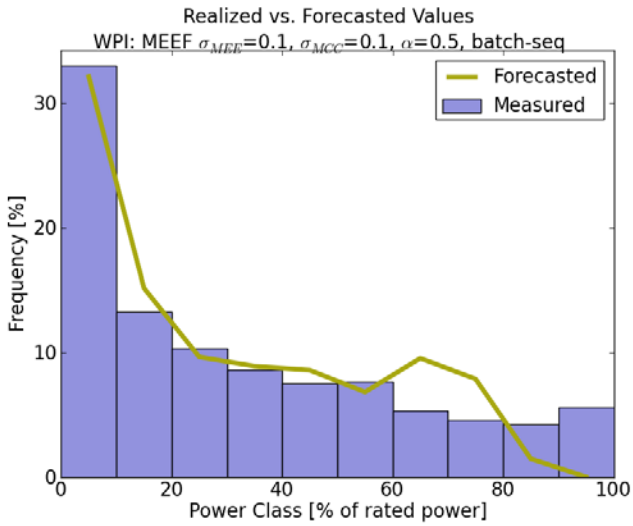

Fig. 2-27 Frequency of occurrence of forecasted and measured values, Wind Farm A – MEEF.

The MEEF, or maximum error entropy with fiducial points, introduces an additional weighting term that regulates the ratio between the MEE and MCC criteria. However, a general conclusion after repeated simulations is that in Wind Farm A, MEEF criteria yield slightly worse results than do MCC criteria. The same conclusion is applicable for the online case. With regard to run time, MEEF is comparable to MEE.

2.4.1.5 Centered Correntropy – cMCC

Figs. 2-28 through 2-31 capture findings related to the cMCC criterion.

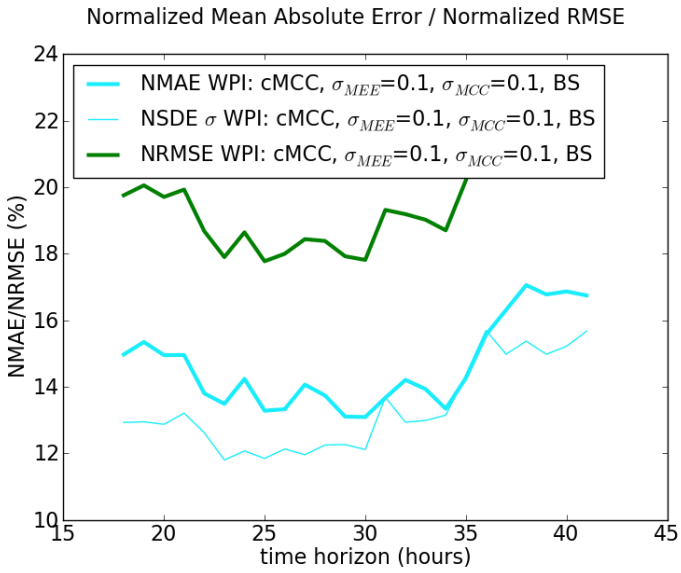

Fig. 2-28 NMAE and NRMSE, offline training, in Wind Farm A – cMCC.

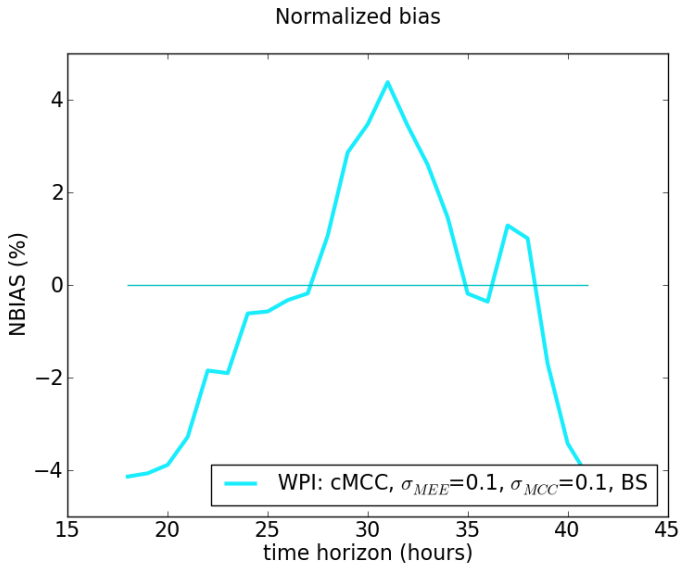

Fig. 2-29 NBIAS for offline training, in Wind Farm A – cMCC.

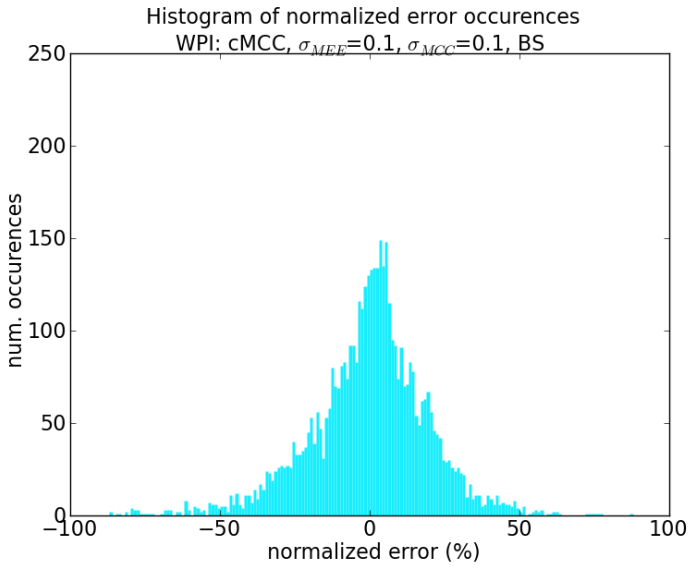

Fig. 2-30 Histogram of error occurrences in Wind Park A – cMCC.

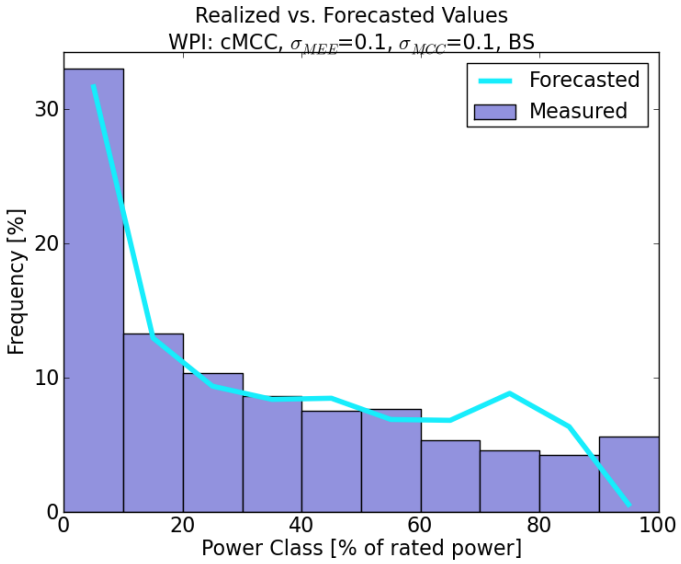

Fig. 2-31 Frequency of occurrence of forecasted and measured values, Wind Farm A – cMCC.

The basic idea behind the cMCC criterion is exploiting the robustness of MEE while not having to suffer from the pronounced bias of the forecast. In this regard, the cMCC behaves as expected, and its bias characteristics are similar to those of MCC.

An interesting observation with regard to cMCC is the use of a learning momentum factor: for WPA training, the learning momentum does not have a desired effect. Instead, when the network is trained with cMCC and learning momentum, the training process has difficulties converging. For this reason in the above and following examples with cMCC, the learning momentum is always set to zero.

2.4.1.6 Summary of Performance for Various Criteria for WFA

This section presents an overview and comparison of the exhibited performance of various criteria for Wind Farm A. For the sake of clarity, the offline measures are compared — in this park, the offline-trained W2P model performs relatively well, and the difference in the "behavior" of different forecasts is slightly more pronounced (Figs. 2-32 and 2-33).

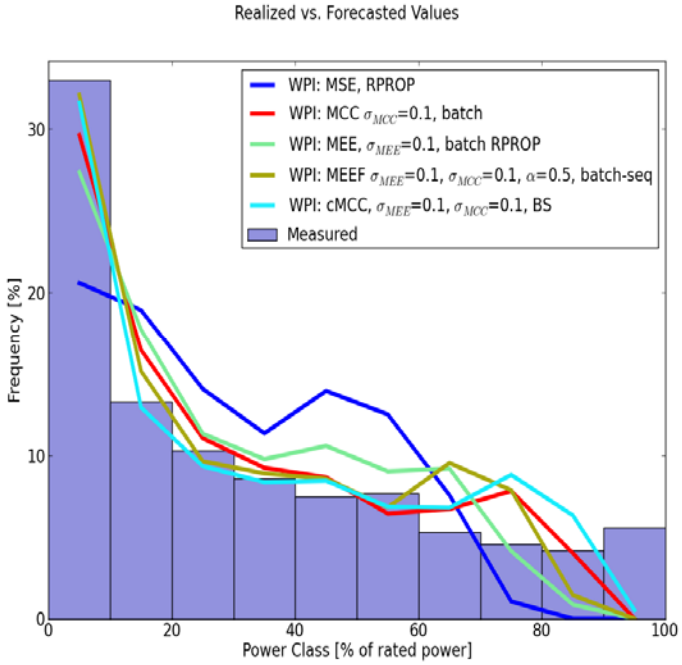

Fig. 2-32 Frequency of occurrence of forecasted and measured values, Wind Farm A – Comparison of performance of various ITL criteria with MSE.

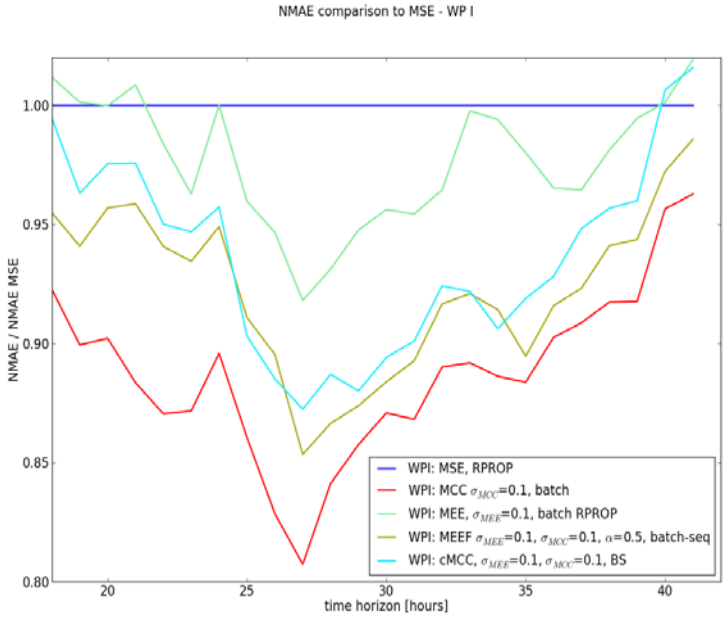

Fig. 2-33 Comparison of NMAE for various ITL criteria with MSE, Wind Farm A.

The latter illustration is particularly interesting: any ITL-based training criterion successfully surpasses the MSE criterion. Furthermore, the best performing criterion for Wind Farm A seems to be MCC, and in the above situation, it manages to achieve almost 20% better performance in terms of NMAE in some hours. Better performance of ITL criteria should not be surprising after considering the error distributions visible in histograms for each of the training methods. The training errors are clearly non-Gaussian — so a measure of error that does not rely on the Gaussian shape of the error curve should perform better.

The former illustration shows the main drawback of using MSE criteria in the WFA setting — it overestimates the mid-range of production values, while underestimating high and low values. ITL methods perform much better in these ranges of values.

All of the forecasting methods seem to exhibit less satisfactory behavior with regard to the highest range of values (close to installed power). The most significant reason for this behavior is described in the following power histograms (Figs. 2-34 and 2-35).

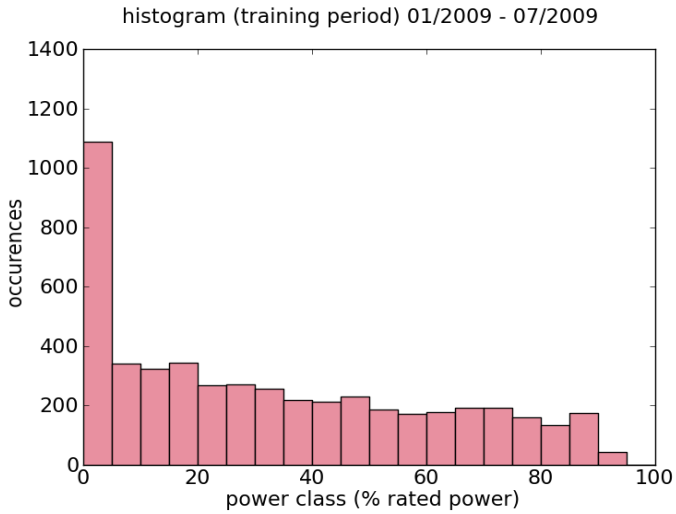

Fig. 2-34 Histogram of power classes for the training period, Wind Farm A.

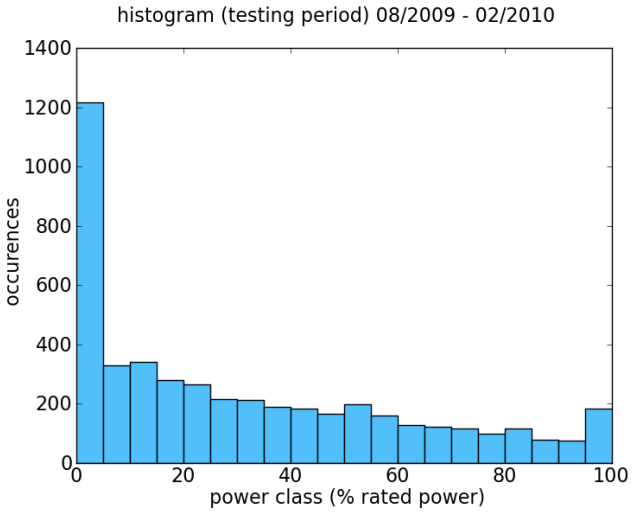

Fig. 2-35 Histogram of power classes for the testing period, Wind Farm A.

Fig. 2-34 presents the histogram for the training period and Fig. 2-35 presents the histogram for the testing period. There are practically no occurrences of highest values in the training dataset, so all of the methods have a difficult task when this class of power has to be predicted. For this reason, the predicted versus realized graphs show that the prediction performance in this power class is somewhat lower. This problem is commonly resolved by using online (adaptive) training methods. However, for the visible impact of using such methods, a longer time span of the dataset used for testing may be required.

One conclusion, however, is clear — ITL methods show a clear advantage over the classic MSE-based training method.

2.4.2 Wind Farm B

This wind park has about half of the installed capacity compared to wind farm A. Even though the two wind parks are in the same area, forecasting WFB represents a slightly more challenging task. The dataset for the WFB represent the wind park's first months of operation, so the frequency of low production is higher. In addition, for such a setting, it is expected that even in a limited duration dataset, the online learning methods provide notably better results than their offline counterparts. All of the general training parameters were the same as for WFA.

2.4.2.1 Minimum Square Error – MSE

For the offline training case (Figs. 2-36 through 2-39), 800 iterations of the iRPROP training algorithm were used. For the iRPROP training algorithm, the learning rate parameter has no effect because it uses an adaptive method of setting the learning rate.

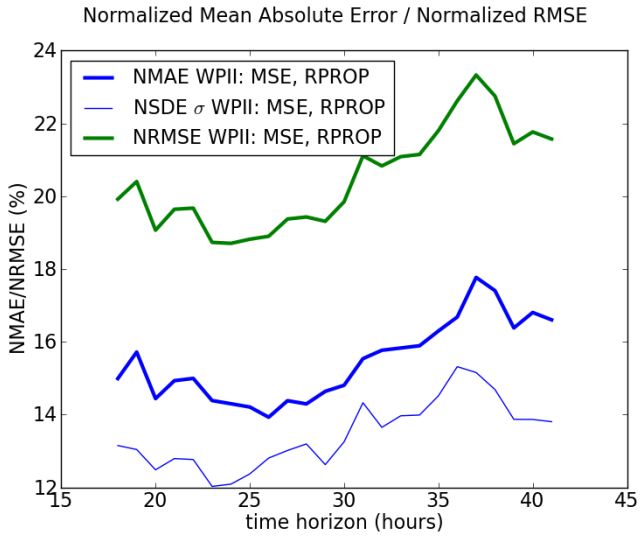

Fig. 2-36 NMAE and NRMSE, for offline training, in Wind Farm B – MSE.

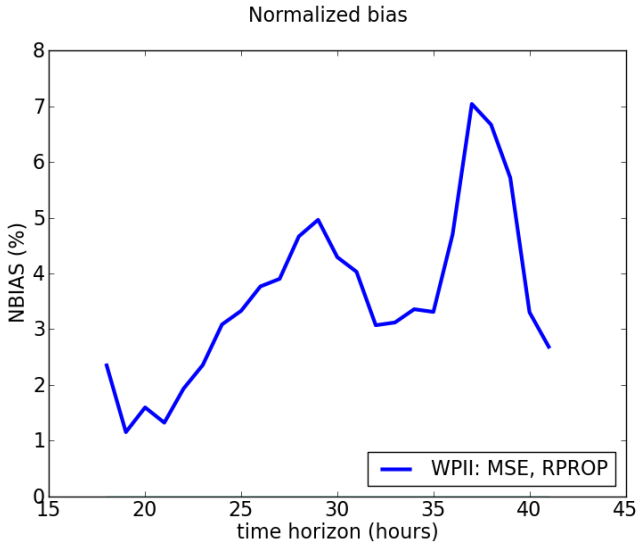

Fig. 2-37 NBIAS for offline training, in Wind Farm B – MSE.

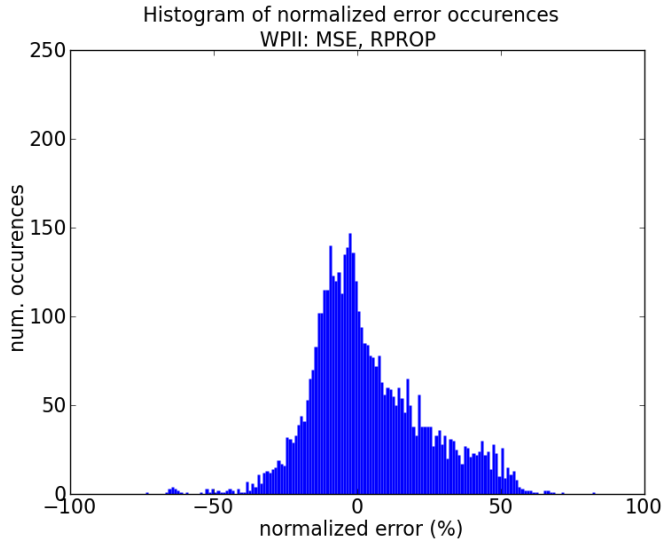

Fig. 2-38 Histogram of error occurrences in Wind Farm B – MSE.

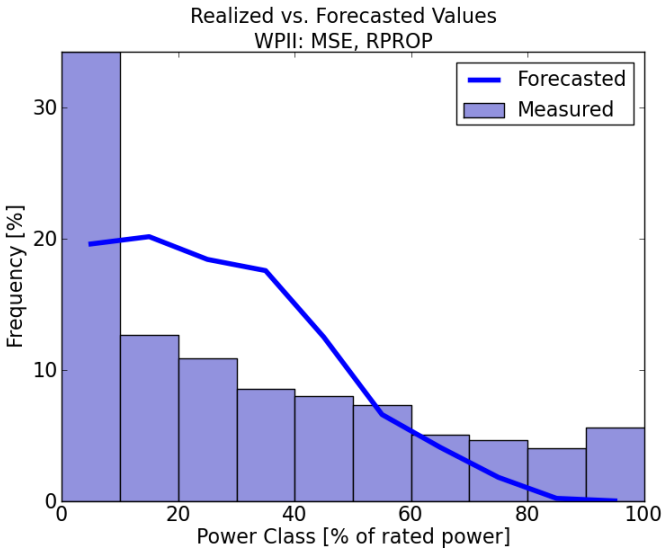

Fig. 2-39 Frequency of occurrence of forecasted and measured values, Wind Farm B – MSE.

In comparison with the WFA results, a significantly larger bias is immediately visible. The fact that Wind Farm B was in the process of establishment also shows the importance of online training. In an online training setting (Figs. 2-40 through 2-42), the learning rate was set to $\eta = 0.001$.

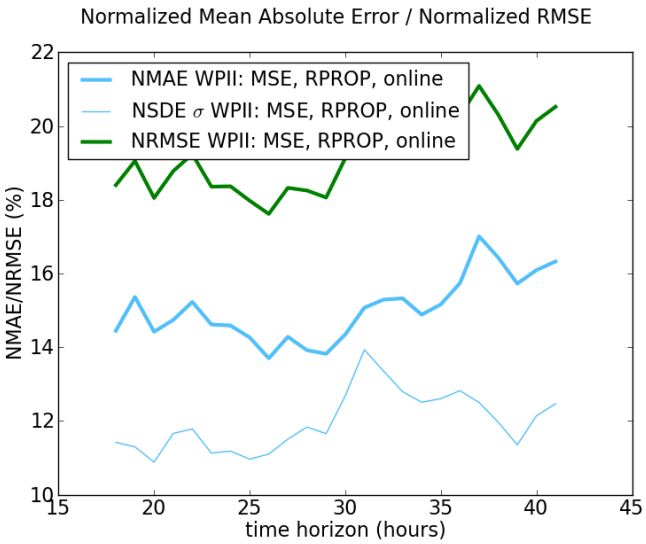

Fig. 2-40 NMAE and NRMSE, for online training, in Wind Farm B – MSE.

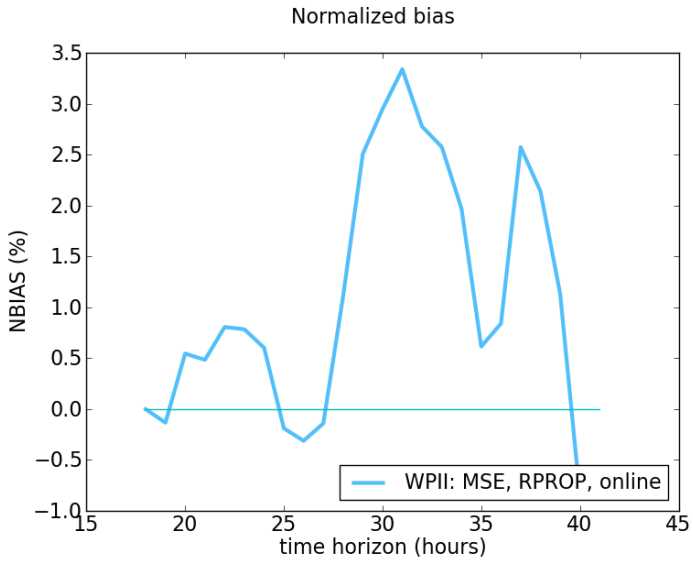

Fig. 2-41 NBIAS for online training, in Wind Farm B – MSE.

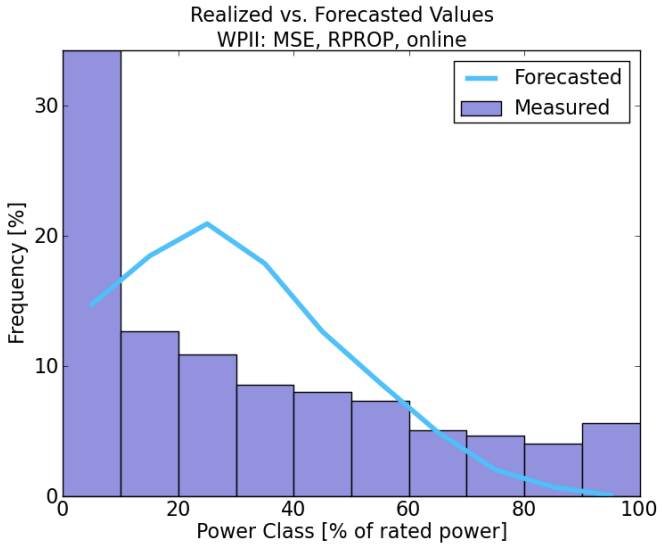

Fig. 2-42 Frequency of occurrence of forecasted and measured values, Wind Farm B – MSE.

The effect of online learning is more pronounced in the case of bias, which is significantly lowered; however, the general characteristic of the forecasted versus realized power distribution is not significantly better when online MSE learning is used.

2.4.2.2 Maximum Correntropy Criterion — MCC

For the MCC criterion, the same settings as in WFA were used: the Gaussian kernel size σ_{MCC} is lowered to 0.1 after the first 200 epochs; and for the remaining 600 epochs, the classic batch backpropagation method is used (Figs. 2-43 through 2-46).

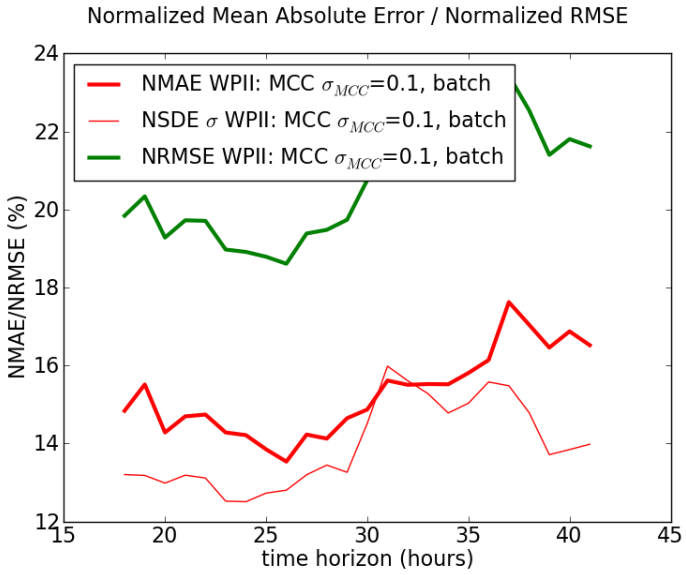

Fig. 2-43 NMAE and NRMSE, for offline training, in Wind Farm B – MCC.

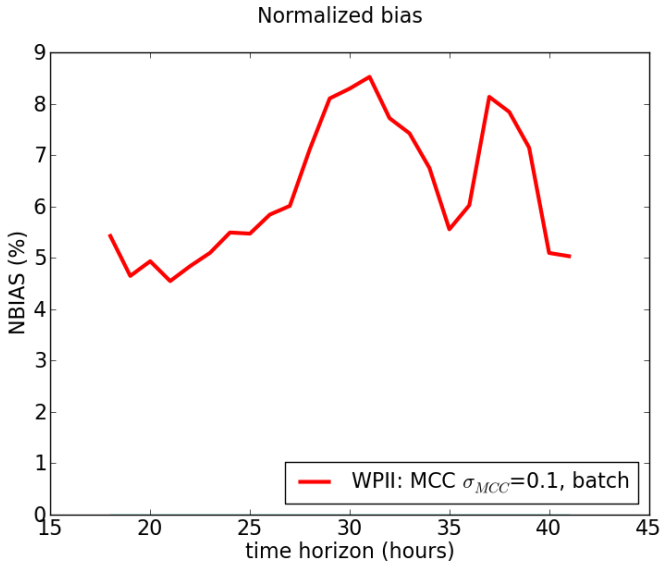

Fig. 2-44 NBIAS for offline training, in Wind Farm B – MCC.

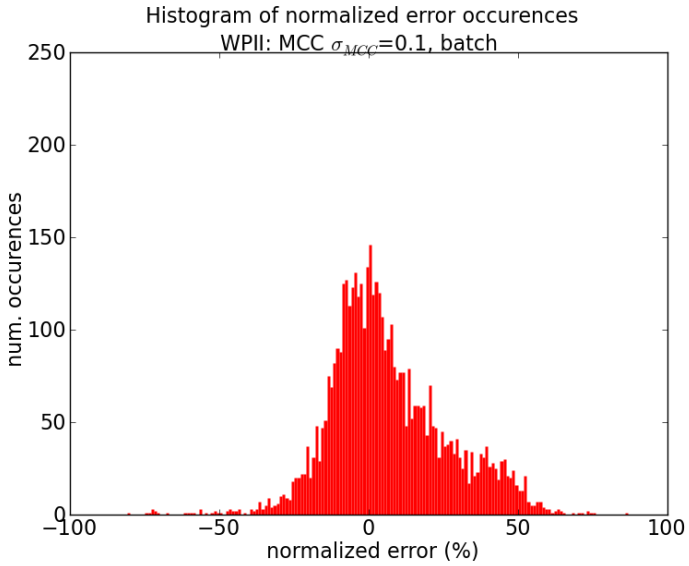

Fig. 2-45 Histogram of error occurrences in Wind Farm B – MCC.

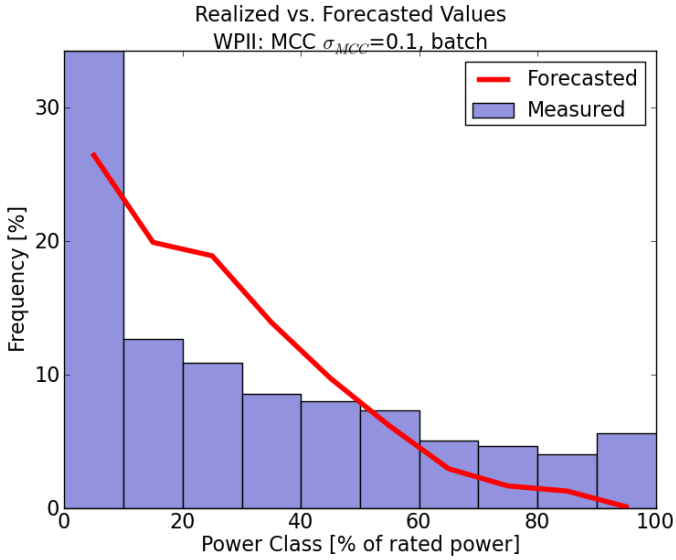

Fig. 2-46 Frequency of occurrence of forecasted and measured values, Wind Farm B – MCC.

The MCC criterion's key trait — that it successfully achieves better prediction for a class below 10% of rated power — is evident here. However, because of the specifics of the power class distribution of WFB, neither MCC is able to accurately follow the shape of the distribution. The improvement in NMAE values, however, is by a lower margin than what is found in the case of WFA.

The addition of online training has a favorable effect in the case of the MCC criterion, too (Figs. 2-47 through 2-49) — both the NMAE and bias are reduced in comparison with the offline MCC method (Fig. 2-43 and Fig. 2-44, respectively).

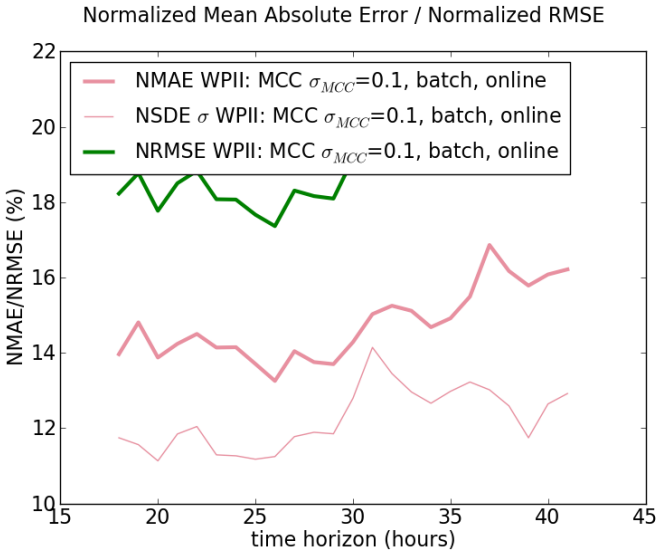

Fig. 2-47 NMAE and NRMSE, for online training, in Wind Farm B – MCC.

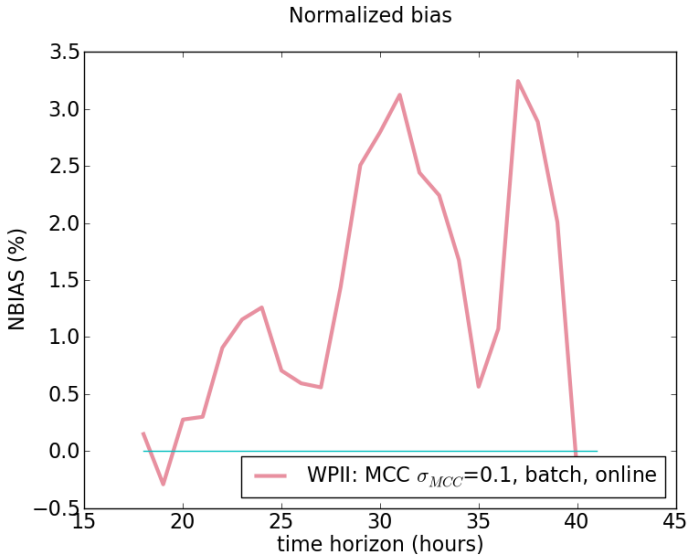

Fig. 2-48 NBIAS for online training, in Wind Farm B – MCC.

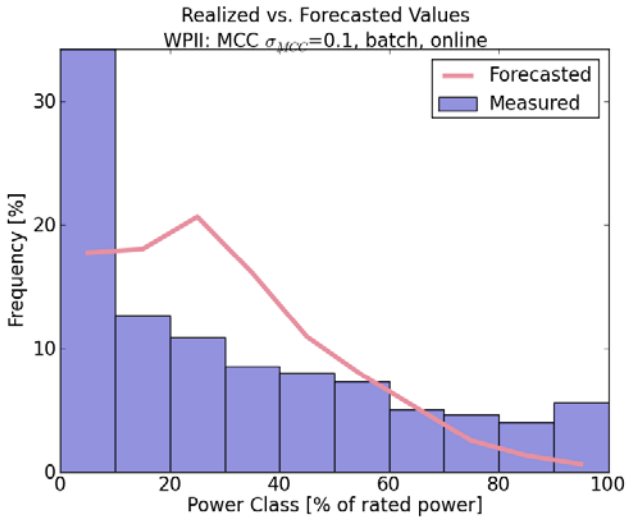

Fig. 2-49 Frequency of occurrence of forecasted and measured values, Wind Farm B – MCC.

2.4.2.3 Minimum Error Entropy Criterion – MEE

Aspects of the MEE criterion's performance are highlighted in Figs. 2-50 through 2-53.

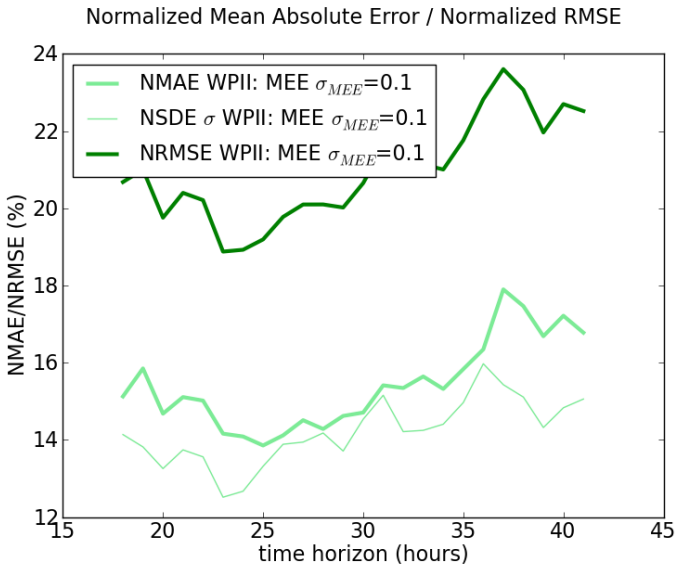

Fig. 2-50 NMAE and NRMSE, offline training, in Wind Farm B – MEE.

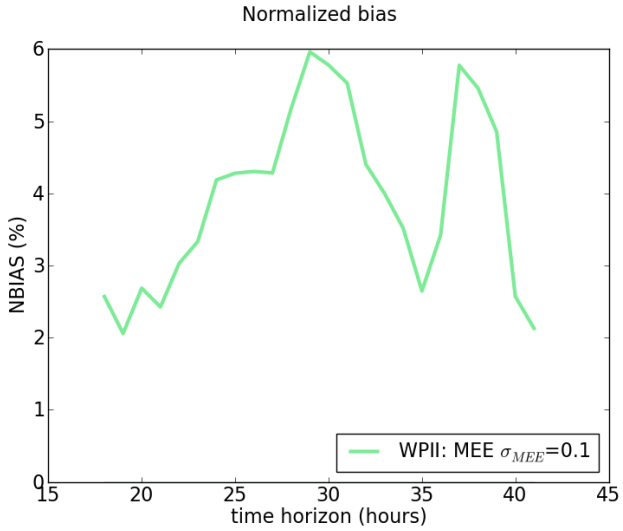

Fig. 2-51 NBIAS for offline training, in Wind Farm B – MEE.

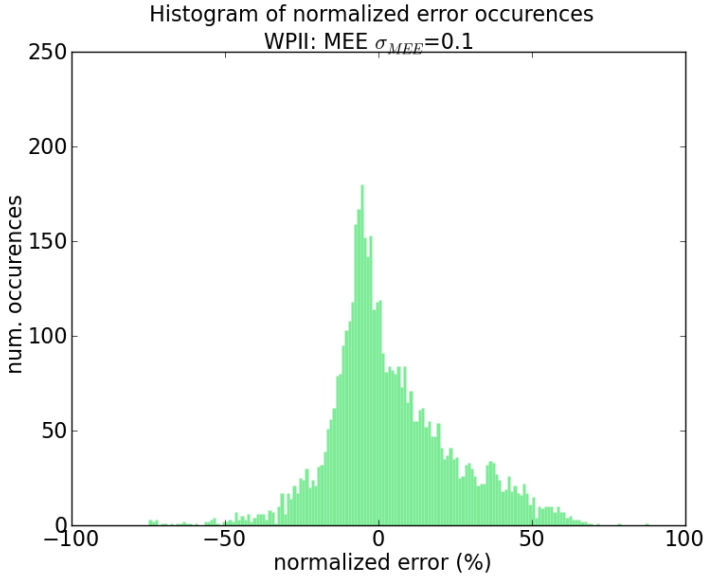

Fig. 2-52 Histogram of error occurrences in Wind Farm B – MEE.

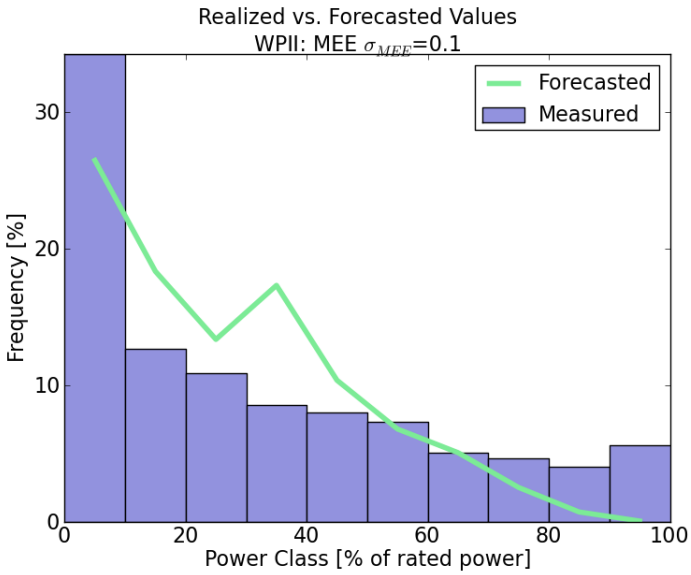

Fig. 2-53 Frequency of occurrence of forecasted and measured values, Wind Farm B – MEE.

The insensitivity of the MEE criterion to mean is shown here, as well, considering that the bias is also relatively large for WFB. However, the MEE criterion seems to be better performing than MCC in terms of following the power class distribution. On the other hand, if the NMAE criterion is used to evaluate predictions, then MCC performs slightly better than MEE.

2.4.2.4 Minimum Error Entropy with Fiducial Points – MEEF

Aspects of the MEEF criterion's performance are highlighted in Figs. 2-54 through 2-57.

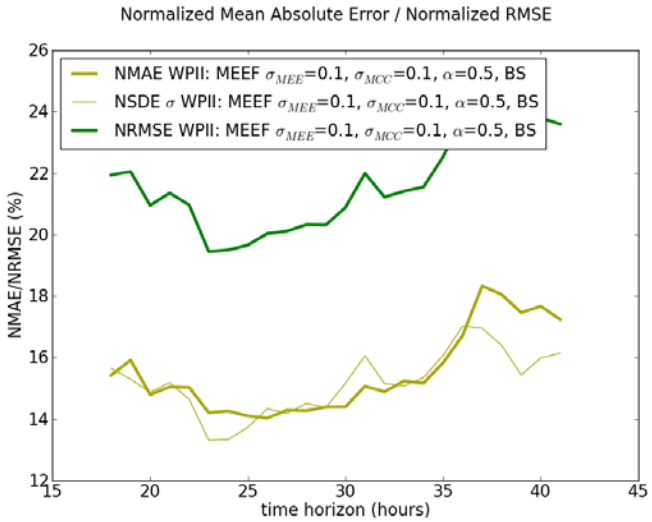

Fig. 2-54 NMAE and NRMSE, offline training, in Wind Farm B – MEEF.

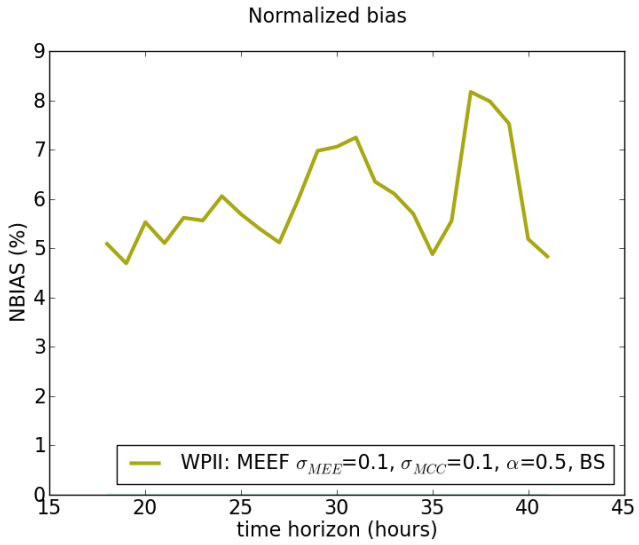

Fig. 2-55 NBIAS for offline training, in Wind Farm B – MEEF.

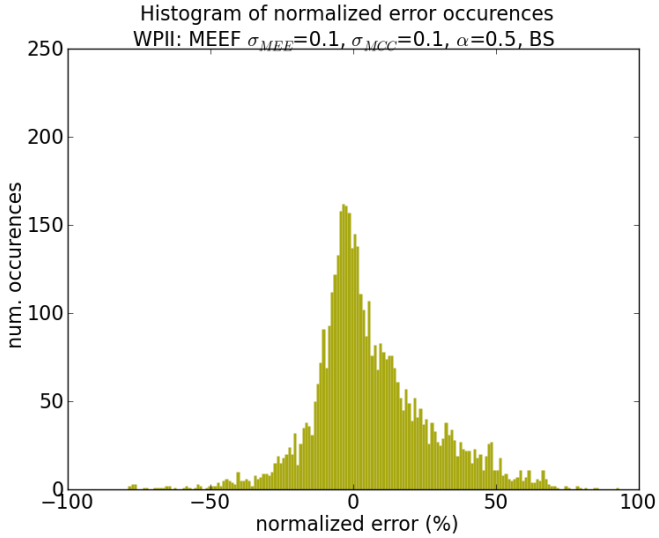

Fig. 2-56 Histogram of error occurrences in Wind Farm B – MEEF.

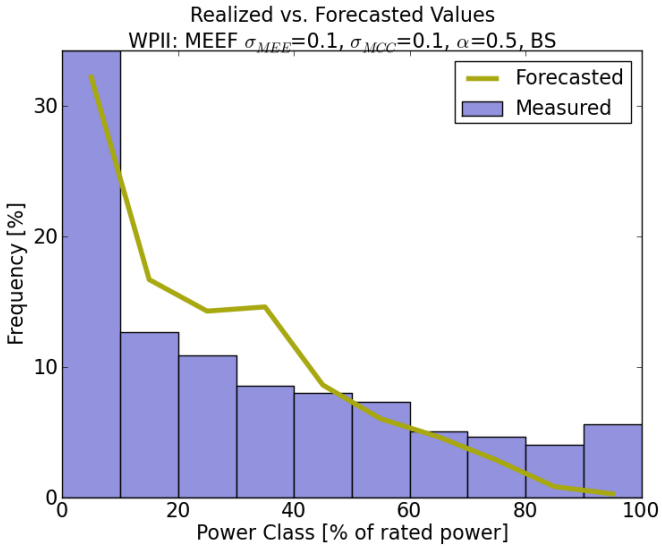

Fig. 2-57 Frequency of occurrence of forecasted and measured values, Wind Farm B – MEEF.

The behavior of MEEF criterion for the case of WFB is fairly similar to that of WFA. In the case of WFB, presumably as a result of the characteristics of the training dataset, the bias of MEEF is notably higher than it is in the case of MEEF applied in WFA. The introduction of online training (Figs. 2-58 through 2-60) is especially visible on the realized versus forecasted distribution graph.

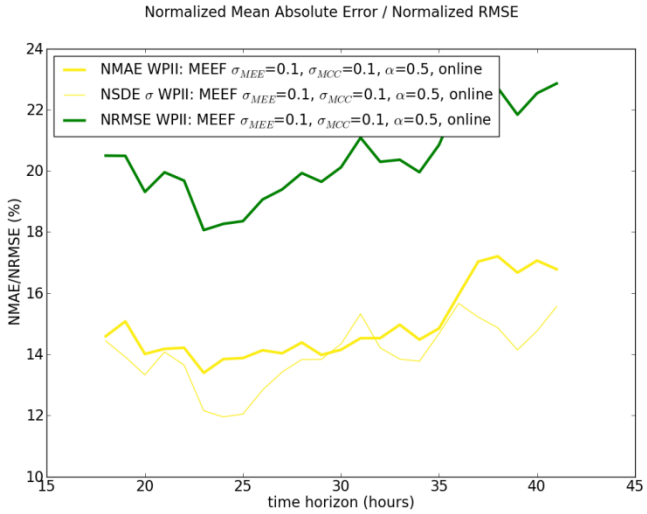

Fig. 2-58 NMAE and NRMSE, online training, in Wind Farm B, MEEF.

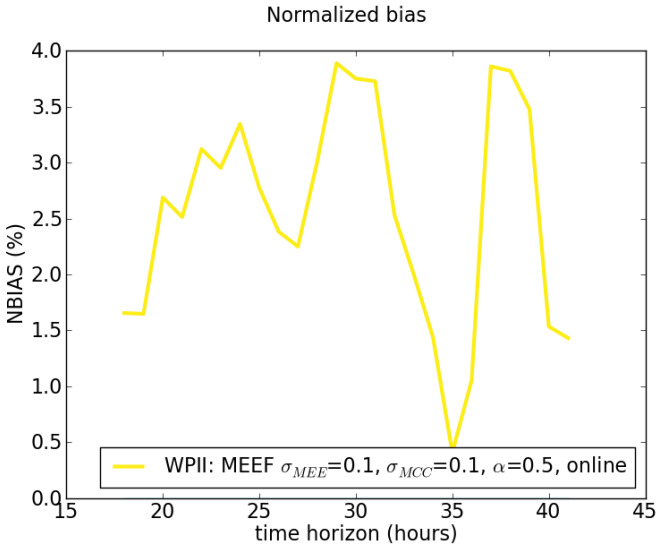

Fig. 2-59 NBIAS for online training, in Wind Farm B, MEEF.

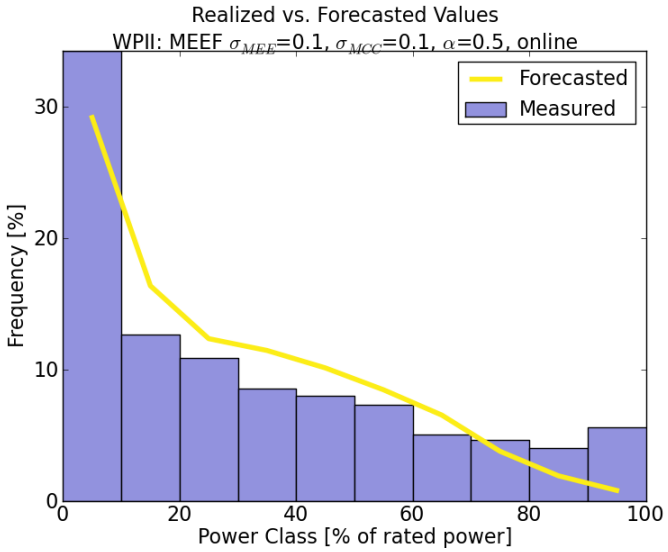

Fig. 2-60 Frequency of occurrence of forecasted and measured values, Wind Farm B.

2.4.2.5 Centered Correntropy – cMCC

The observation for WFB (Figs. 2-61 through 2-67) again confirms the value of online learning in the WFB dataset — the online learning performs notably better. For instance, the bias of cMCC online is half the bias of cMCC offline.

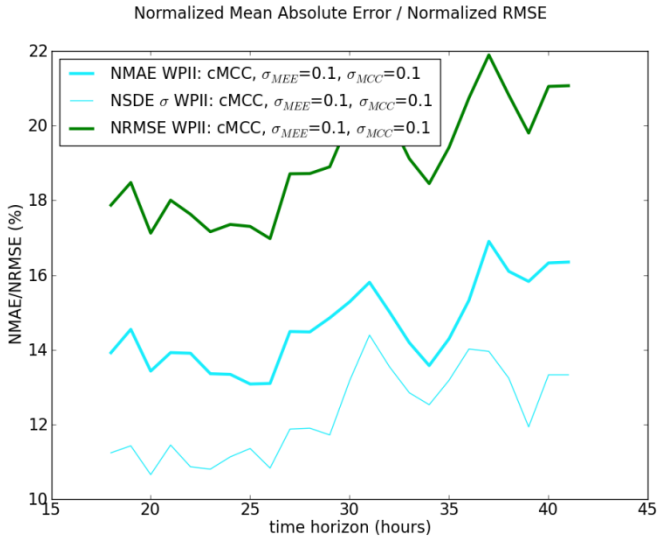

Fig. 2-61 NMAE and NRMSE, offline training, in Wind Farm B – cMCC.

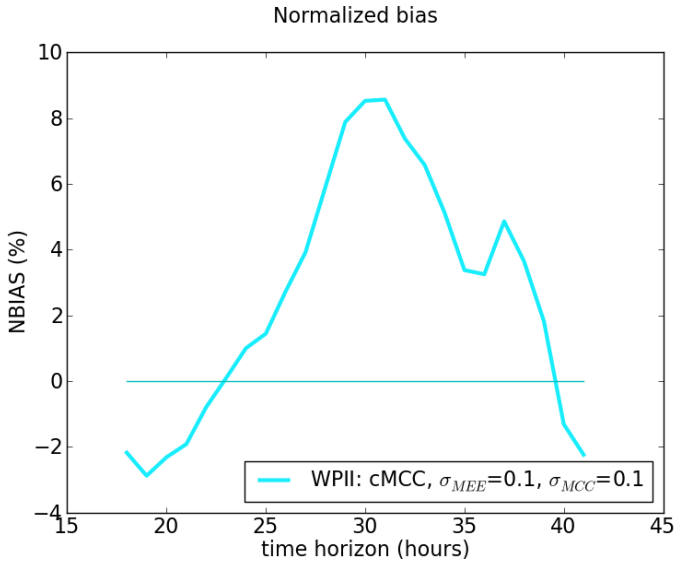

Fig. 2-62 NBIAS for offline training, in Wind Farm B – cMCC.

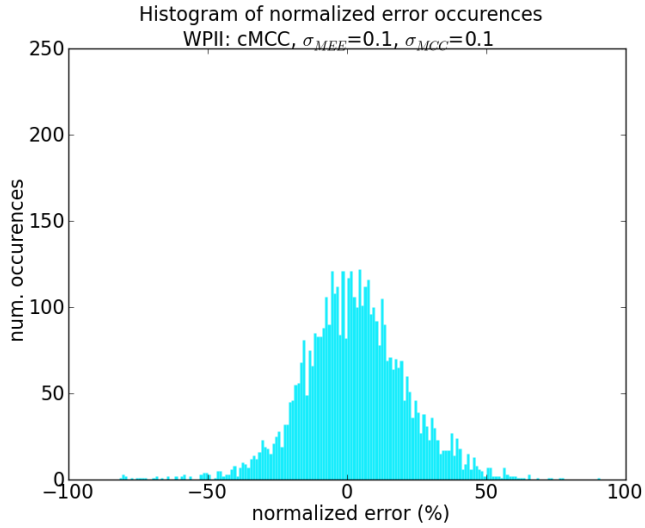

Fig. 2-63 Histogram of error occurrences in Wind Farm B – cMCC.

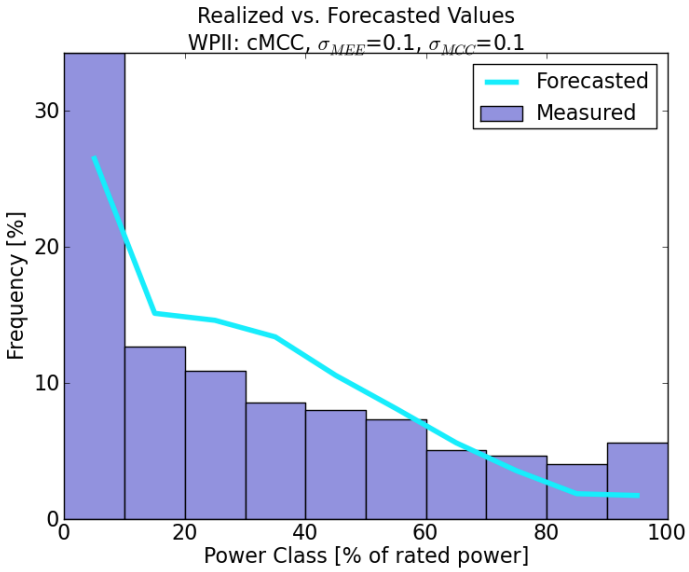

Fig. 2-64 Frequency of occurrence of forecasted and measured values, Wind Farm B – cMCC.

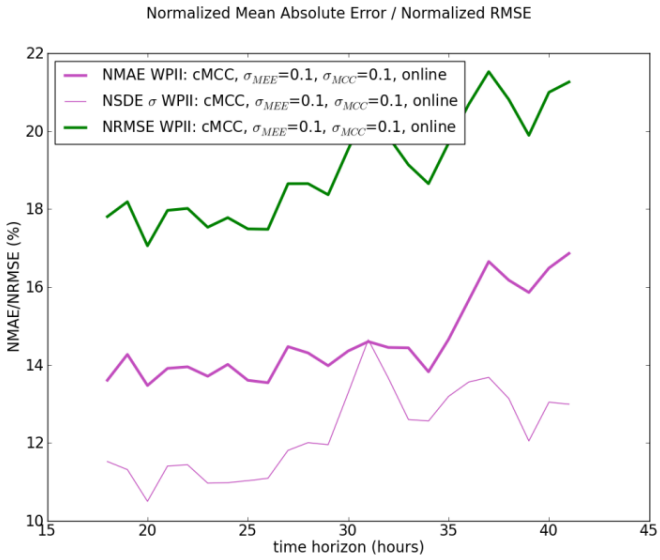

Fig. 2-65 NMAE and NRMSE, online training, in Wind Farm B – cMCC.

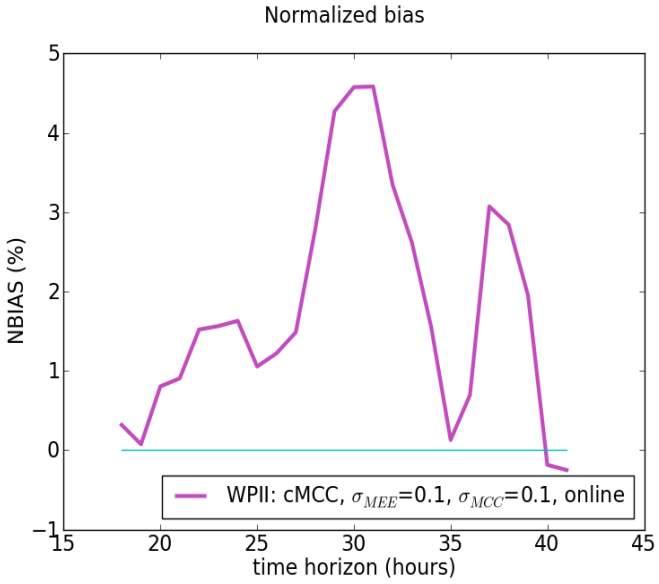

Fig. 2-66 NBIAS for online training, in Wind Farm B – cMCC.

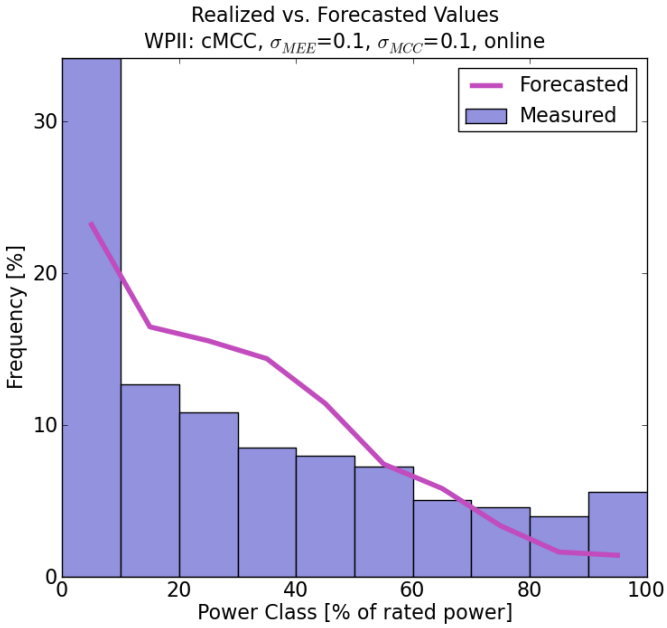

Fig. 2-67 Frequency of occurrence of forecasted and measured values, Wind Farm B – cMCC.

2.4.2.6 Summary of Performance for Various Criteria for WFB

This section presents an overview and comparison of the exhibited performance for various criteria at Wind Farm B. Summary results are presented in Figs. 2-68 through 2-71.

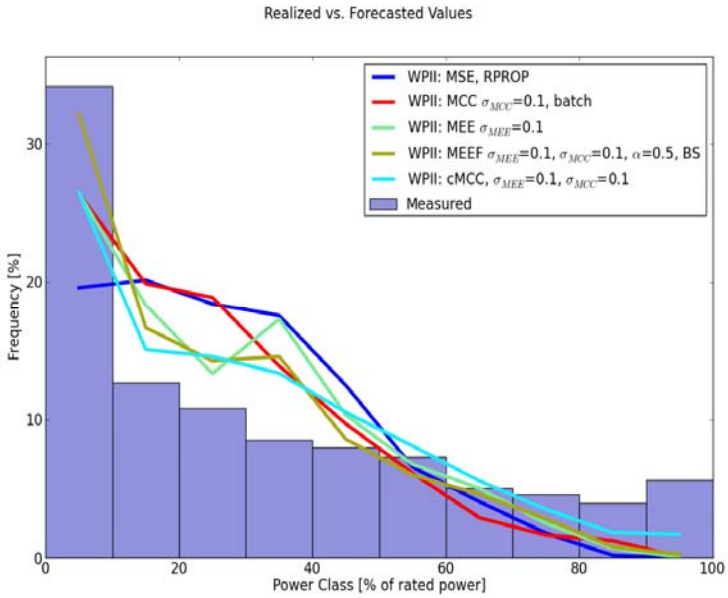

Fig. 2-68 Frequency of occurrence of forecasted and measured values, Wind Farm B – Comparison of performance of various ITL criteria with MSE.

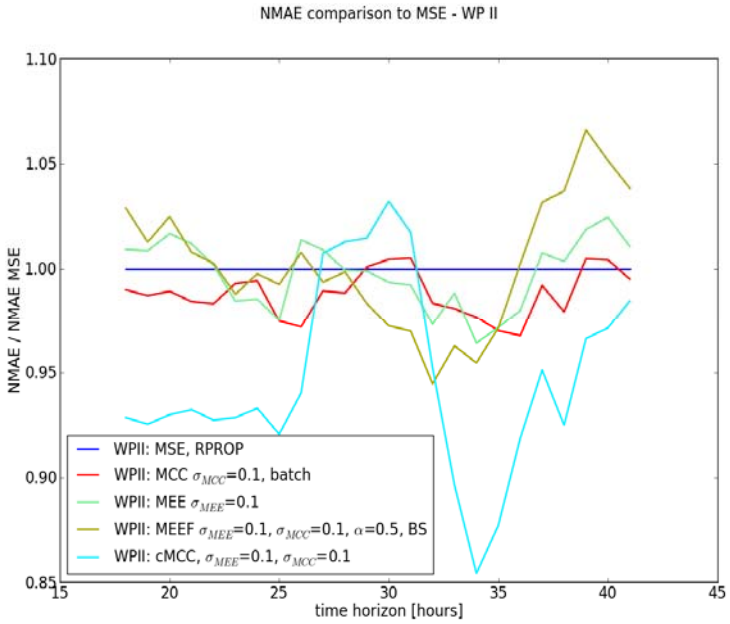

Fig. 2-69 Comparison of NMAE for various ITL criteria with MSE, Wind Farm B.

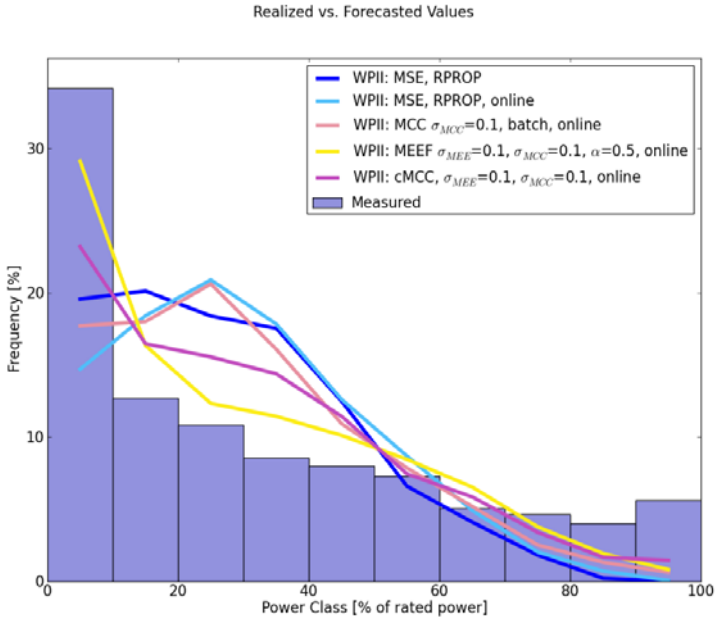

Fig. 2-70 Frequency of occurrence of forecasted and measured values, Wind Farm B – Comparison of performance of various online ITL criteria with MSE.

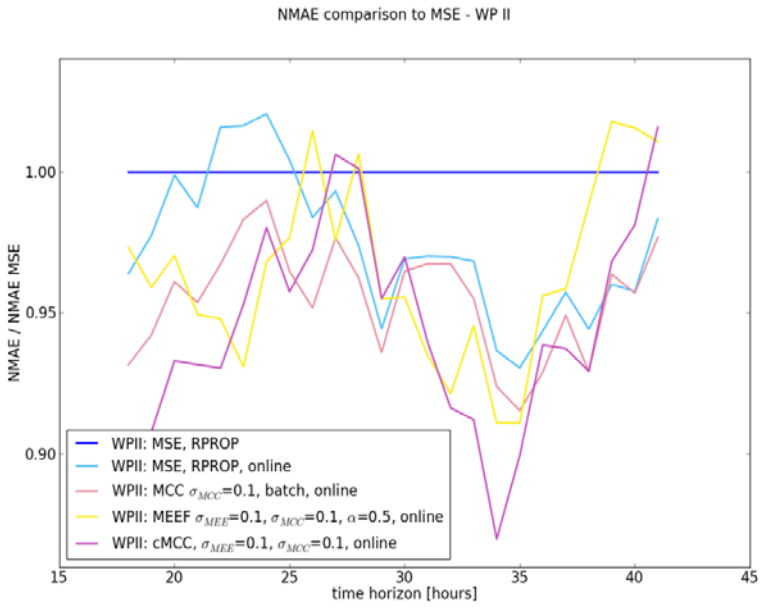

Fig. 2-71 Comparison of NMAE for various online ITL criteria with MSE, Wind Farm B.

As compared to the figures above, it is obvious that the influence of online training is larger than the differences among the training criteria. Still, MCC with the online setting exhibits favorable performance, especially considering the small computation burden it requires. An interesting observation is performance of "pure" MEE criterion when compared to MSE in an offline setting — the MEE's insensitivity to mean and corresponding problems — are a possible reason for this effect.

2.5 Conclusions

The results presented in this chapter document the application of ITL training criteria to WPF for two large-scale wind farms located in the Midwest, namely, Wind Farm A and Wind Farm B.

Prior experience with the use of ITL criteria has shown favorable performance with distinct advantages over the classic MSE criterion. These findings have been confirmed here, and a new ITL-based criterion, centered correntropy, is introduced in this chapter.

With regard to location specifics, the two wind farms are located next to each other and therefore experience similar terrain and weather conditions; however, their behavior and data differ. The turbines in WFA have been operated for a longer period of time, and the data series used for training showed a significant degree of dependence on the training criterion applied. On the other hand, WFB had just been established at the beginning of the observed data series, and thus we found that the use of online training was significantly more important.

In theory, online training should assure better performance in the presence of concept drift. This effect, however, may not be very pronounced if the dataset contains a limited amount of data, so an elaborated concept drift is not visible in the data. In this chapter, slightly more than a year's worth of data was available for both training and testing. For an established wind farm, the effect

of online training in the treatment of concept drift may appear to be more pronounced if the W2P model is tested over a longer period of time. Thus, for more detailed testing of the online training methods, a more extensive dataset would provide better insight into the behavior of online ITL criteria.

3 NEW CONTRIBUTIONS TO WIND POWER UNCERTAINTY FORECASTING: KERNEL DENSITY FORECAST

This chapter is organized as follows: After an introductory description, Section 3.2 presents the motivation to represent the uncertainty by density functions. Section 3.3 describes two algorithms for kernel density forecast: the classical Nadaraya-Watson estimator and the quantile-copula estimator. Section 3.4 presents the results for two case studies: National Renewable Energy Laboratory's (NREL's) Eastern Wind Integration and Transmission Study (EWITS) dataset and a U.S. wind farm. Section 3.5 presents a discussion about the goodness in probabilistic forecasts, and finally, Section 3.6 presents the conclusions and next steps.

3.1 Introduction

A single-valued forecast (or point/deterministic forecast) cannot provide to the forecast user information on the dispersion of observations around the predicted value. Therefore, it is essential to generate, together with (or as an alternative to) point forecasts, a representation of the wind power uncertainty.

The algorithms from the state-of-the-art in wind power uncertainty forecasting can be found in [1]. Three key features have captured researchers' attention: (i) how to represent the wind power uncertainty; (ii) the model chain for uncertainty forecasting; and (iii) time-adaptive (or online) models to cope with non-stationary data.

The wind power uncertainty can take the form of probabilistic forecasts, risk indices, or scenarios for short-term wind power generation. Probabilistic forecasting consists of expressing the wind power generation or forecast error in "probabilistic terms," such as: (a) parametric representation (e.g., Gaussian distribution); (b) moments of the distributions (e.g., mean, standard deviation, skewness); (c) a set of quantiles; (d) probability mass function (pmf); and (e) probability density function (*pdf*). Normally, the uncertainty representation is determined by the algorithm used (e.g., if quantile regression is used, the uncertainty is represented by a set of quantiles).

The traditional model chain for wind power uncertainty forecasting, according to Juban *et al.* [14], consists of using as input the forecast errors or point forecasts from a wind power deterministic forecasting model. The uncertainty estimation model is placed after the model that produces wind power deterministic forecasts. One example of this model chain combines adapted resampling with fuzzy inference, as developed by Pinson [15]. A preferred approach consists either of using the Numerical Weather Prediction (NWP) forecast error as input for the uncertainty estimation method or computing the uncertainty directly from the NWP points. This class of algorithms avoids an intermediate step (conversion of wind to power, the W2P step) because they can produce probabilistic and deterministic forecasts. For instance, the local quantile regression described by Bremnes [16] forecasts the wind power generation quantiles based on information about explanatory variables (e.g., NWP forecasts); a set of quantiles characterize the uncertainty, and the point forecast could be associated with the median (quantile 50%). The Kernel Density Estimation described by Juban *et al.* [17] also provides an uncertainty estimation and point forecast.

Most of the methods available in the literature are models trained in an offline mode, with the models unable to cope with changes in the underlying distributions of the several variables. Examples of offline forecasting algorithms are the quantile regression presented by Bremnes [16] and the model described by Juban *et al.* [17]. On the other hand, the tendency in the state of the art is to develop algorithms capable of adapting to changes in data; some examples are the time-adaptive quantile regression model described by Møller *et al.* [18] and the conditional parametric autoregressive model recently developed by Pinson [19].

Consequently, an algorithm for wind power uncertainty forecasting shall ideally have the following as prerequisites: (i) a high level of flexibility to represent wind power uncertainty; (ii) time-adaptive characteristics; and (iii) ability to avoid an intermediate step to compute point forecasts. However, point forecasts from different models could represent additional and useful information for uncertainty forecast.

In this chapter, new contributions for the advancement beyond the state-of-the-art in wind power forecast uncertainty are presented. Two algorithms for wind power density forecast, which respect the three prerequisites mentioned above, are described.

3.2 Motivation to Represent Wind Power Uncertainty by Probability Density Functions

From an information theory perspective, the *pdf* contains all of the information associated with a random variable. For instance, it enables computation of the moments of the forecasted distribution. Therefore, we may argue that the *pdf* representation is generic and can be transformed into several uncertainty forms, such as quantiles, standard deviation, or skewness.

The best way to represent uncertainty is determined by the end user's requests and the nature of the decision-making problem being addressed. In general, one cannot talk about better and worse uncertainty representations, only of more or less adequate representations. However, the *pdf* delivers the necessary flexibility for addressing several decision-making problems.

The problem of finding the "optimal" wind power bidding for the electricity market can be formulated with different methods when wind power uncertainty is considered. When the objective is to maximize the expected profit (or minimize the expected cost of imbalances), the aim consists of finding the optimal quantile, which for some electricity markets is determined by imbalances in price ratios [20]. It is possible to extract the optimal quantile from the *pdf* for each hour and, consequently, the "optimal" decision under the expected value paradigm. Botterud *et al.* [21] presented an approach based on maximizing the utility; for this approach, the *pdf* enables the production of a probability mass function (*pmf*) that can be used to compute the expected utility. Bourry *et al.* [22] described an approach based on portfolio theory where a trade-off between expected income and risk (described by the conditional value-at-risk) is evaluated to find the "optimal" bid. This approach is in line with the work developed by Matos [23], where the aim is to describe uncertainty by a set of deterministic risk measures. For this problem, the knowledge of the *pdf* allows the computation of any risk measure. For instance, it is possible to evaluate a trade-off between expected income and risk described by the variance and skewness.

The *pdf* representation is also useful to set the required operating reserve for the current and next days using, for instance, the method presented by Matos and Bessa [24]. The *pdf* representation provides the full probability distribution, which allows a better characterization of the tails. According to Bessa and Matos [25], the tails in the operating reserve problem are the critical factor, in particular if the system operator (SO) prefers a higher level of security (e.g., loss of load probability around 1%).

The method described by Pinson *et al.* [26] to represent the uncertainty by scenarios with temporal correlation of forecast errors could also benefit from the *pdf* representation. With a forecasted *pdf*, the distribution is fully characterized and there is no need to perform an exponential interpolation.

According to Nielsen *et al.* [27], the quantiles may cross in quantile regression, because to compute each quantile it is necessary to solve an independent optimization problem. The *pdf* forecasts supply directly non-crossing quantiles. However, this detail is minor because this behavior means that the uncertainty is lower between the quantiles.

Finally, and as mentioned by several authors [17][28]–[30], for multimodal distributions a density forecast allows computation of the modes instead of just computing the expected value (which, in this case, is not a good summary of the distribution). It is unlikely to find wind power multimodal density distributions, but the mode or the median is still a better deterministic forecast, because normally the wind power distributions are highly skewed. With this approach, it is possible to follow the method described by Faugeras [30]:

> *One can consider the statistician should first estimate the full conditional distribution to fully quantify the input of X on Y and then, once the general shape of the conditional density is given, to build some sensible point predictors and predictive sets. This is especially relevant if the predictive distribution is multi-modal or skewed, which often arises in applications with non-Gaussian or non-linear phenomena.*

3.3 Kernel Density Forecasting Methodology

3.3.1 Basic Concepts

The theoretical framework for Kernel Density Forecasting (KDF) is the Kernel Density Estimation (KDE) and Conditional Kernel Density Estimation (CKDE).

3.3.1.1 Kernel Density Estimation

KDE, which was introduced by Rosenblatt [31] in 1956, with several properties derived by Parzen [32] in 1962, consists of using a non-parametric estimator of a density function. Given independent and identically distributed data (i.i.d) $X_1,\ldots,X_n$ drawn from an unknown density function *f*, the univariate KDE is given by:

$$\hat{f}_X(x) = \frac{1}{N \cdot h} \cdot \sum_{i=1}^{N} K\left(\frac{x - X_i}{h}\right) \tag{3-1}$$

where N is the number of samples, K is a Kernel function, and h the bandwidth parameter. Wolverton *et al.* [33] presented an alternative in 1959 formulated as:

$$\hat{f}_X(x) = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{1}{h_i} \cdot K\left(\frac{x - X_i}{h}\right) \tag{3-2}$$

where h_i is the bandwidth parameter for each sample i. For instance, if the Kernel function is a Gaussian, the following estimator is considered:

$$\hat{f}_X(x) = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{1}{\sqrt{2 \cdot \pi}} e^{-\frac{(x - X_i)^2}{2 \cdot h^2}} \tag{3-3}$$

The previous equation consists of placing a Gaussian Kernel in each sample X_i. The corresponding density function results, depicted in Fig. 3-1, show a density distribution estimated by dividing by 8 the sum of eight Gaussians (with variance 0.09) centered on eight samples.

Given i.i.d multivariate data $X_{1d},\ldots,X_{2d}$ from d different variables drawn from an unknown multivariate density function f, the multivariate KDE is given by:

$$\hat{f}(x_1, \ldots, x_d) = \frac{1}{N \cdot h_1, \ldots, h_d} \cdot \sum_{i=1}^{N} K\left(\frac{x_1 - X_{i1}}{h_1}, \ldots, \frac{x_d - X_{id}}{h_d}\right) \tag{3-4}$$

where K is a multivariate Kernel function and $h_1,\ldots,h_d$ a bandwidth vector.

When the support of x (range of possible values) is different for d variables (which is the case of the wind power problem), the approach consists in using the product kernel estimator [34]:

$$\hat{f}(x_1, \ldots, x_d) = \frac{1}{N} \cdot \sum_{i=1}^{N} \prod_{j=1}^{d} K_j\left(\frac{x_j - X_{ij}}{h_j}\right) \tag{3-5}$$

where K_j is the kernel function for variable j with bandwidth h_j.

Fig. 3-2 depicts the joint *pdf* computed using (3-5) for data from a real wind farm. This *pdf* represents the probability density associated to each point plotted in the wind speed versus wind power scatter of Fig. 3-3. The region with highest density in Fig. 3-2 matches the zones with highest concentration of points in Fig. 3-3.

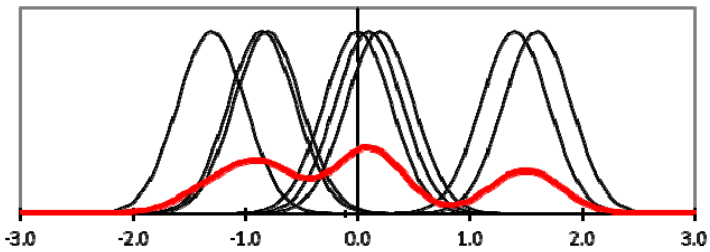

Fig. 3-1 Illustration of the Parzen window method to estimate the pdf from a sample of 8 points D = {-1.3; -0.85; -0.8; 0; 0.1; 0.2; 1.4; 1.6} and with h=0,3. In red: the estimated pdf, obtained after the division by 8 of the sum of the individual Gaussians, so that its integral is equal to 1.

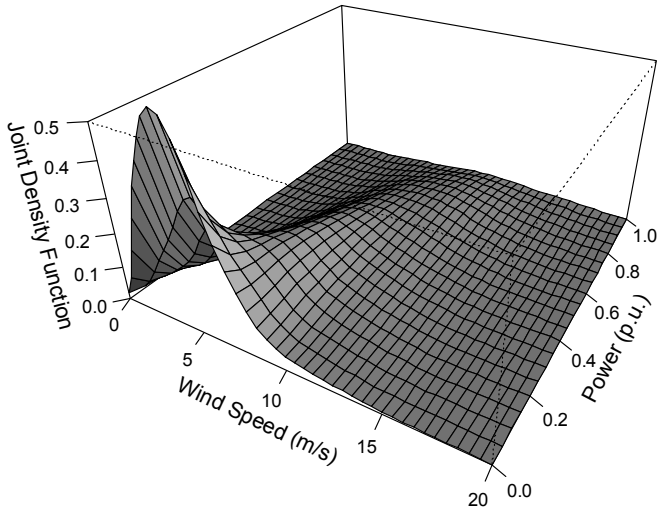

Fig. 3-2 Joint probability density function of forecasted wind speed and measured wind power.

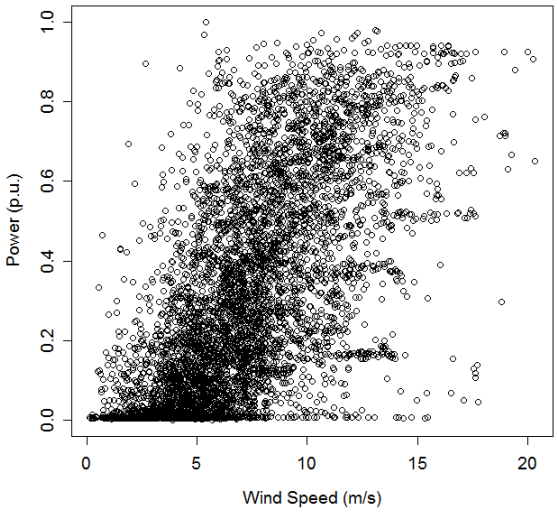

Fig. 3-3 Scatter plot of forecasted wind speed versus measured wind power.

3.3.1.2 Conditional Kernel Density Estimation

Conditional density estimation consists of estimating the density of a random variable Y, knowing that the explanatory random variable X is equal to x. In other words, it consists of estimating the density of Y conditioned to $X=x$, $f(y|X=x)$. The conditional density can be formulated as follows:

$$\frac{\qquad}{\qquad} \tag{3-6}$$

where $f(x,y)$ is the multivariate density function of X and Y (joint distribution function), and $f(x)$ is the marginal density of X.

It is also possible to have nonparametric conditional density estimation. The classic approach is the Nadaraya-Watson kernel smoother proposed by Rosenblatt [35] in 1969:

$$\hat{f}(y|X=x) = \frac{\hat{f}_{XY}(x,y)}{\hat{f}_X(x)} = \sum_{i=1}^{N} K_h(y - Y_i) \cdot w_i(x) \quad (3\text{-}7)$$

where $w_i(x) = \frac{K_h(x-X_i)}{\sum_{i=1}^{N} K_h(x-X_i)}$.

3.3.2 Nadaraya-Watson Estimator

Hyndman *et al.* [36] considered the following modified Nadaraya-Watson (NW) estimator:

$$\hat{f}(y|X=x) = \sum_{i=1}^{N} K_{h_y}(y - Y_i) \cdot w_i(x) \quad (3\text{-}8)$$

where $w_i(x) = \frac{K_{h_x}(x-X_i)}{\sum_{i=1}^{N} K_{h_x}(x-X_i)}$

and where the bandwidth h_y controls the smoothness of each conditional density in the y direction, while h_x controls the smoothness between conditional densities in the x direction.

This estimator will be considered for wind power density forecast because its implementation is simple, and implementing its time-adaptive version is straightforward. Moreover, this estimator is also useful for benchmarking with the quantile-copula (QC) approach that will be described in the next sub-section.

3.3.3 Quantile-Copula Estimator

The quantile-copula estimator was introduced by Faugeras [37]. According to the authors, its main advantages over the existing methods are that: the methods based on the NW estimator are numerically unstable when the denominator is close to zero; for a problem with several explanatory variables, this method has only one kernel product, instead of two; at a conceptual level, density estimation should only be based on density estimation methods and not on regression approaches (like the NW estimator). Moreover, this estimator easily allows the inclusion of bounded data, such as the wind power.

The main difference from the NW estimator is in the joint density function of Y and X. Almost at the same time, Faugeras [37] and Bouezmarni and Rombouts [38] proposed the idea of using a copula for modeling the dependency structure between Y and X. Regarding copulas, the Sklar theorem [39] says the following for the bivariate case:

Let H be a two-dimensional distribution function with marginal distribution functions F and G. Then there is a copula C such that

$$H(x,y) = C\big(F(x), G(y)\big). \quad (3\text{-}9)$$

Conversely, for any univariate distribution functions F and G and any copula C, the function H is a two-dimensional distribution function with marginals F and G. Furthermore, if F and G are continuous, then C is unique.

This theorem means that the multivariate distribution function can be separated into two parts: (i) marginal functions that can be estimated separately; and (ii) a dependency structure between

the marginal which is modeled by the copula. For more details about copulas, see Nelson [40]. A conditional density estimator can be built from (3-9). So, we know that:

$$F_{XY}(x,y) = C\big(F_X(x), F_Y(y)\big) \tag{3-10}$$

then (3-5) can be replaced by

$$f(x,y) = \frac{\partial^2}{\partial u \cdot \partial v} \cdot C(u,v) = f_X(x) \cdot f_Y(y) \cdot c(u,v) \tag{3-11}$$

where u and v are a quantile transform of the data, $u=F_X(x)$ and $v=F_Y(y)$, and c is the copula density function.

Replacing (3-11) in (3-6), we have the following conditional density estimator:

$$f(y|X=x) = f_Y(y) \cdot c(u,v) \tag{3-12}$$

Now, it is necessary to build an estimator for (3-12). The idea proposed by Bouezmarni and Rombouts [38] was a semiparametric approach, where a parametric model is considered for the copula, and the marginal distributions are represented by a nonparametric model (empirical distribution function). However, we followed the idea described by Faugeras [37], where the copula density is estimated with KDE.

The estimator for $f_Y(y)$ is the KDE in (3-1). The copula density estimator is the estimator in (3-5) as follows (for the bivariate case):

$$\hat{c}(u,v) = \frac{1}{N} \cdot \sum_{i=1}^{N} K\left(\frac{u-U_i}{h}\right) \cdot K\left(\frac{v-V_i}{h}\right) \tag{3-13}$$

where U_i and V_i are the data transformed by the empirical cumulative distribution function — $U_i=F_X^e(X_i)$ and $V_i=F_Y^e(Y_i)$. An empirical cumulative distribution function (cdf) is defined as:

$$F^e(t) = \frac{1}{N} \cdot \sum_{i=1}^{N} \mathrm{I}(x_i \leq t) \tag{3-14}$$

where I is the indicator function of event $x_i \leq t$.

Fig. 3-4 depicts the copula *pdf* computed with (3-13) for the quantile transform of the wind speed and wind power, using (3-14), from a real wind farm. This copula density function represents the probability density associated to each point plotted in the wind speed versus the wind power scatter of Fig. 3-3.

The copula represents the dependence structure between the two variables. Therefore, from Fig. 3-4 and Fig. 3-5, we see that there is a strong dependence in the two extreme corners, for example, when there are lower quantiles in wind speed (lower wind speed values), the wind power quantiles also present lower values with a higher probability.

An interesting conclusion is that this copula density seems very similar to a family of parametric copulas, the elliptical copulas [41].

The quantile-copula CKDE is written as:

$$\hat{f}(y|X=x) = \frac{1}{N \cdot h} \cdot \sum_{i=1}^{N} K_0\left(\frac{y-Y_i}{h}\right) \cdot \frac{1}{N} \cdot \sum_{i=1}^{N} K_1\left(\frac{F_X^e(u)-F_X^e(U_i)}{h_q}\right) \cdot K_2\left(\frac{F_X^e(v)-F_X^e(V_i)}{h_q}\right) \quad (3\text{-}15)$$

3.3.4 Formulation of the Wind Power Density Forecast Problem

The wind power density forecast problem can be formulated as: forecast the wind power *pdf* at time step t for each look-ahead time step $t+k$ of a given time horizon (e.g., up to 72 hours ahead) when knowing a set of explanatory variables (e.g., NWP forecasts, wind power measured values, hour of the day).

Translating this sentence to an equation, we have:

$$\hat{f}_P\left(p_{t+k}|X=x_{t+k|t}\right) = \frac{f_{P,X}\left(p_{t+k},x_{t+k|t}\right)}{f_X\left(x_{t+k|t}\right)} \quad (3\text{-}16)$$

where p_{t+k} is the wind power forecasted for look-ahead time $t+k$, and $x_{t+k|t}$ are the explanatory variables forecasted for look-ahead time step $t+k$ and available/launched at time step t.

Equation (3-16) can be solved using the approaches presented in Sections 3.3.2 and 3.3.3, where the variable Y is the wind power, and the explanatory variables X are for instance: NWP variables (wind speed, wind direction, pressure), wind power point forecast, and measured wind power.

Fig. 3-6 depicts what is called in [36] a stacked conditional plot. This plot represents the information contained in (3-16) and can be obtained with both approaches presented in Sections 3.3.2 and 3.3.3. It enables one to see the changes in the wind power density function conditioned to different values of forecasted wind speed (ranging from 0 to 20 m/s). The conditional densities for intermediate values of wind speed are very broad, and we may also detect a higher concentration of density in the tails for lower and higher values of wind speed.

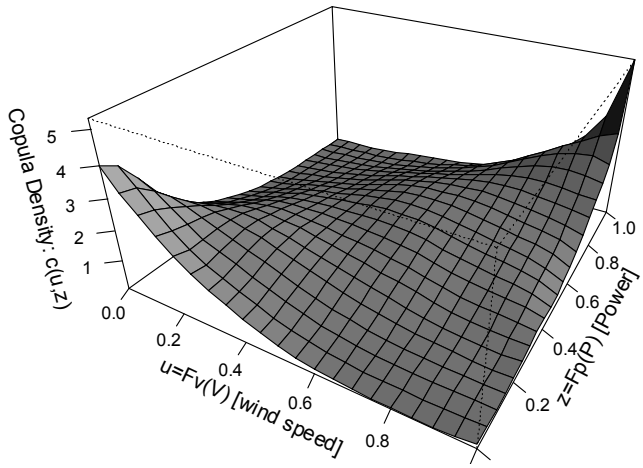

Fig. 3-4 Bivariate copula density function of forecasted wind speed and measured wind power.

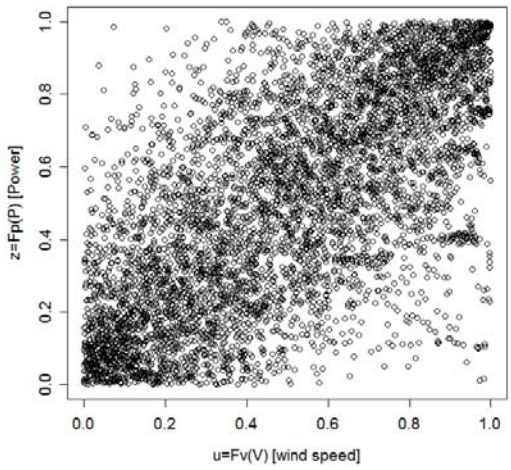

Fig. 3-5 Scatter plot of quantile transform of forecasted wind speed versus measured wind power.

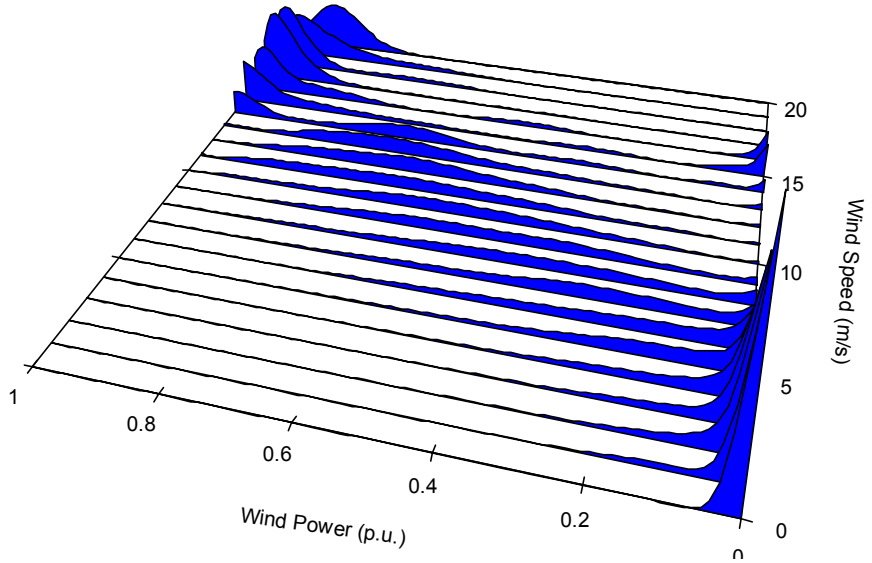

Fig. 3-6 Stacked conditional plot for wind power and wind speed.

3.3.5 Kernel Function Choice

The choice of the kernel function for the wind power forecasting problem is a critical issue. It depends on the type of variable and data and represents an enormous impact on the model performance.

In regard to the data type, we have in the wind power problem four different types: (i) wind power bounded between 0 (e.g., zero generation) and 1 (e.g., rated power); (ii) wind speed bounded between 0 and +Inf; (iii) variables (such as temperature) between –Inf and + Inf; and

(iv) circular variables, such as the hour of the day and the wind direction. For these four types, different kernels should be considered.

In the literature, several kernels were proposed for variables with support [0,1], where some of these kernels are as follows: two different beta kernels proposed by Chen [42]; a boundary kernel developed by Zhang and Karunamuni [43]; the swapped Chen beta kernel proposed by Jones and Henderson [44]; and the bivariate Gaussian copula developed by Jones and Henderson [44].

We have tested all of these kernels, and for the wind power problem, the two Chen beta kernels presented better results than did the others. Zhang [45] compared the performance of the Chen estimators with this boundary kernel, and the main conclusion was that for densities not exhibiting a shoulder (*f'(0)=0*), the beta kernel estimators have a serious boundary problem, and their performances are inferior to the boundary kernel. However, the results show that wind power problem respects the shoulder condition.

Moreover, the variables *u* and *v* in (3-15) are bounded between [0,1]; therefore, the Chen beta kernels will be used for these variables.

The two Chen beta kernels considered for modeling the wind power are:

$$\hat{f}_1(x) = \frac{1}{N}\sum_{i=1}^{N} K_{x/b+1,(1-x)/b+1}(X_i) \tag{3-17}$$

where $K_{p,q}$ is the density function of a Beta(p,q) random variable, with p and q as the two positive shape parameters and with b being the bandwidth parameter of $K_{\mathrm{p,q}}$; and

$$\hat{f}_2(x) = \frac{1}{N}\sum_{i=1}^{N} K^*_{x,b}(X_i) \tag{3-18}$$

$K_{x,b}^{*}$ are boundary kernels defined as

$$K^*_{x,b}(t) = \begin{cases} K_{x/b,(1-x)/b}(t) \;\; if\; x \in [2b, 1-2b] \\ K_{\rho(x),(1-x)/b}(t) \;\; if\; x \in [0,2b) \\ K_{x/b,\rho(1-x)}(t) \;\; if\; x \in (1-2b, 1] \end{cases} \tag{3-19}$$

where $\rho(x,b) = 2b^2 + 2.5 - \sqrt{4b^4 + 6b^2 + 2.25 - x^2 - x/b}$ and $K_{p,q}$ is a Beta(p,q) density function.

Fig. 3-7 and Fig. 3-8 depict the beta kernel shape for five different points and the Gaussian kernel for the same points, respectively. As shown, the beta kernels present a varying shape according to the values of *x*; in fact, the varying shape changes the amount of smoothing applied to the kernel estimator. Moreover, the kernels (and consequently the estimator) are non-negative, in contrast to the Gaussian kernel. The Gaussian kernel shape is fixed for any value of *x*. The Gaussian kernel may lead to inconsistent results at the boundaries.

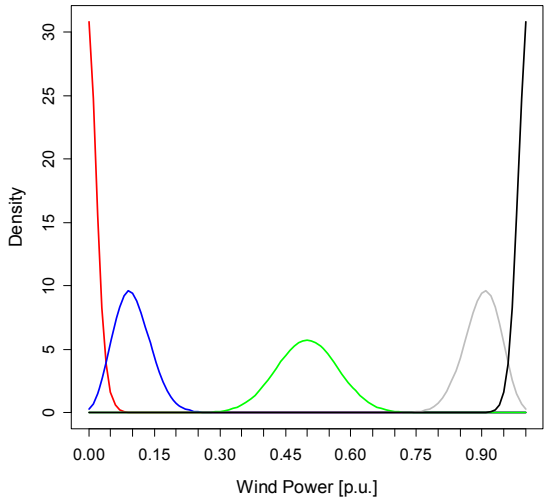

Fig. 3-7 Beta kernels of (3-17) for b=0.02 (red [x=0.01], blue [x=0.1], green [x=0.5], grey [x=0.9], black [x=0.99]).

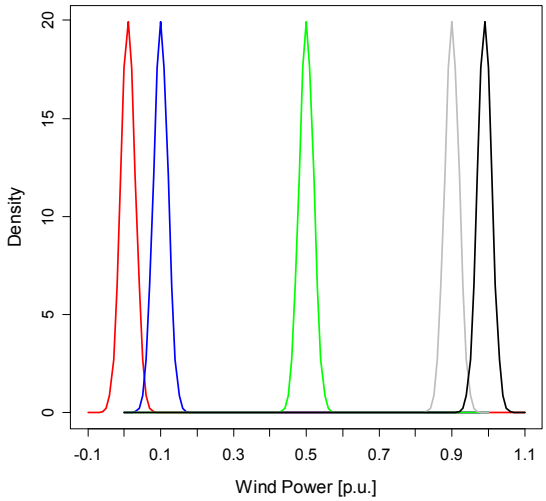

Fig. 3-8 Gaussian kernels of (3-3) for h=0.02 (red [x=0.01], blue [x=0.1], green [x=0.5], grey [x=0.9], black [x=0.99]).

As mentioned by Gourieroux and Monfort [46], the integrals computed from the beta kernels may not converge to their theoretical counterpart. This result may lead to distributions that do not have an integral (are of the distribution) equal to 1; in other words, this method leads to distributions that have no unit mass. Moreover, the kernel is also inconsistent for distributions that are point mass at 0% and 100%. This result is attributable to a lack of normalization, and the idea proposed by Gourieroux and Monfort is a modified beta kernel estimator (named "macro-beta"):

$$\hat{f}'(x) = \frac{\hat{f}(x)}{\int_0^1 \hat{f}(x)dx} \tag{3-20}$$

Because this formulation represents only a change of scale, the normalization for the CKDE is employed over the conditional function of (3-16).

For the variables with support [0,+Inf], the kernels proposed in the literature are two gamma kernels proposed by Chen [47] and the boundary kernel proposed by Zhang [48]. The tests performed for the wind power problem resulted in better performance for both of Chen's gamma kernels.

The two gamma kernels considered for modeling the wind speed are:

$$\hat{f}_1(x) = \frac{1}{N}\sum_{i=1}^{N} K_{x/b+1,b}(X_i) \tag{3-21}$$

where b is the bandwidth parameter of $K_{p,q}$, which is the density function of a Gamma(p,q) random variable with p as the shape parameter and q as the scale parameter; and

$$\hat{f}_2(x) = \frac{1}{N}\sum_{i=1}^{N} K_{\rho_b(x),b}(X_i) \tag{3-22}$$

where $K_{p,q}$ is a Gamma(p,q) density function and $\rho_b(x)$ is given by

$$\rho_b\ (x) = \begin{cases} x/b \quad if \quad x \geq 2b \\ \frac{1}{4}(x/B)^2 + 1 \quad if \quad x \in \ [0,2b) \end{cases} \tag{3-23}$$

For variables with unbounded support, the natural choices are the Gaussian kernel or the bi-weight kernel.

Finally, circular variables are a particular case for KDE. For instance, the difference between a wind direction of 350° and 10° is only 20°; the Euclidian distance (represented as ||.||) is 340°. The approach is to use circular distributions, such as the wrapped normal distribution or the von Mises distribution [49]. In this case, and because it is mathematically more simple and a close approximation to the wrapped normal distribution, we used the von Mises distribution. The von Mises distribution is given by:

$$g(\theta; \mu, \kappa) = \frac{1}{2\pi \cdot I_0(\kappa)} e^{\kappa \cdot cos(\theta - \mu)} \tag{3-24}$$

where I_0 is the modified Bessel function of the first kind and order 0 and defined by

$$I_0(\kappa) = \frac{1}{2\pi}\int_0^{2\pi} e^{\kappa \cdot cos(\theta)} d\theta \tag{3-25}$$

The parameter μ is the directional center of the distribution, κ is the concentration parameter, and θ belongs to any interval of length 2π. The concentration parameter can be used to control the degree of smoothing in circular KDE, and it is analogous to the bandwidth parameter — although larger values lead to less smoothing. Fig. 3-9 depicts an example of circular KDE for the wind direction data. The points are represented in polar coordinates and placed in the circle line, while the five colored lines represent the density estimation for these sample points.

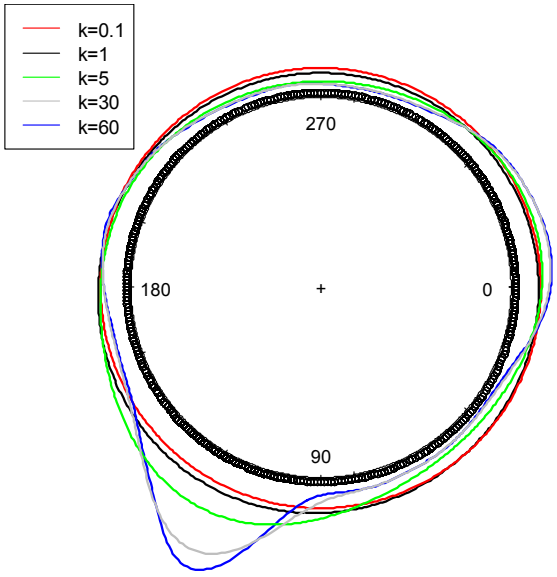

Fig. 3-9 Circular kernel density estimation for the wind direction data.

Note that the circular kernels must also be used for variables *u* and *v* in the quantile-copula approach. In this case, it is necessary to perform a change of scale from [0,1] to [0, 6.266] (in radians).

3.3.6 Time-adaptive Estimator

Wegman and Davies [50] introduced a recursive estimator of KDE for (3-2) in 1979. The density function can be calculated recursively using the following:

$$\hat{f}_n(x) = \frac{n-1}{n} \cdot \hat{f}_{n-1}(x) + \frac{1}{n \cdot h_i} \cdot K\left(\frac{x - X_i}{h_i}\right) \tag{3-26}$$

The extension to the multivariate case (3-4) is straightforward.

Equation (3-26) allows updating of the density function when new samples are available without also forcing the need to recompute the entire density function. However, as the number of *t* increases, the ratio *(n-1)/n* approaches one (and 1/n approaches zero), and then the new samples become redundant. Moreover, if there is a change in the generating structure of the data (non-stationary data), this recursive estimation is incapable of automatically discarding older data.

In order to overcome these problems, Wegman and Marchette [51] proposed the KDE estimator with exponential smoothing. The basic idea of exponential smoothing consists of the following:

$$Y_t = \sum_{i=0}^{\infty} (1-\lambda) \cdot \lambda^{(i)} \cdot X_{t-i}^k, \; 0 < \lambda < 1 \tag{3-27}$$

where λ is a constant between 0 and 1, and X^k_t is the k^{th} power of the random variable X at time t. This expression may be reformulated in a recursive formula as:

$$Y_t = (1-\lambda)\cdot X_t^k + \lambda \cdot Y_{t-1} \tag{3-28}$$

Suppose that X_t has stationary moments so that the expected value is $E[X_t^k] = E[X^k]$. Then, $E[Y_t] = \sum_{i=0}^{\infty}(1-\lambda)\cdot\lambda^{(i)}\cdot E\left[X_{t-i}^k\right] = E[X^k]$. This formulation means that in stationary cases, the exponentially smoothed Y_t has exactly the same expectation as X^k would have.

Note that λ (called the forgetting factor) controls how quickly or slowly the exponential smoothing adapts to the new data (exponential forgetting). A value of λ close to one means that the exponential smoothing puts more weight on the historical data and little weight on the most recent values, whereas when λ is closer to zero, the opposite is true.

Wegman and Davies [50] applied this exponential smoothing to the recursive KDE of (3-26), and the KDE formulation with adjustable discarding of old data becomes:

$$\hat{f}_n(x) = \lambda \cdot \hat{f}_{n-1}(x) + \frac{(1-\lambda)}{h_i}\cdot K\left(\frac{x-X_i}{h_i}\right) \tag{3-29}$$

Note that λ replaces *(n-1)/t* and *(1-λ)* replaces *1/n*, and its value should be slightly below one. Caudle and Wegman [52] mentioned that λ can be represented in terms of n, and thus we have:

$$\lambda = \frac{n}{n+1} \tag{3-30}$$

So, n corresponds to the size of the equivalent sliding window in the time-adaptive KDE.

To show the effect of the forgetting factor in contrast to the recursive estimation in KDE, an estimate of the density was constructed using 1,000 points from an exponential distribution (rate equal to 3) and then 500 points from a Normal Dist (mean=0,std. dev=1). The density estimation was evaluated and updated in another 500 points sampled from Normal Dist (mean=0,std. dev=1). This treatment introduces an artificial (and abrupt) change in the data structure in order to simulate what is known in the literature as "concept drift."

The estimated density for the 500 points of the test dataset is depicted in Fig. 3-10 for: batch estimation, where the density was computed from the 2,000 points; recursive estimation using the 500 points from the test dataset and (3-26); and exponential smoothing with λ=0.995 and using (3-29). The Gaussian Kernel was used. As shown in Fig. 3-10, the recursive formula and the batch estimation are numerically identical, and since they are unable to forget historical data, their density estimation does not correspond to the current data structure. On other hand, the exponential smoothing is capable of learning from the new examples; consequently, the density estimated with this method follows more precisely the histogram of the 500 points from the test dataset.

Fig. 3-11 depicts the density estimated with four different values of λ. A higher value of λ (very close to one) is incapable of adapting to the new data structure, whereas for lower values of λ

(0.9 in this case), the method becomes numerically unstable. Therefore, the value of λ should be a trade-off between the degree of non-stationarity and the numerical stability of the KDE.

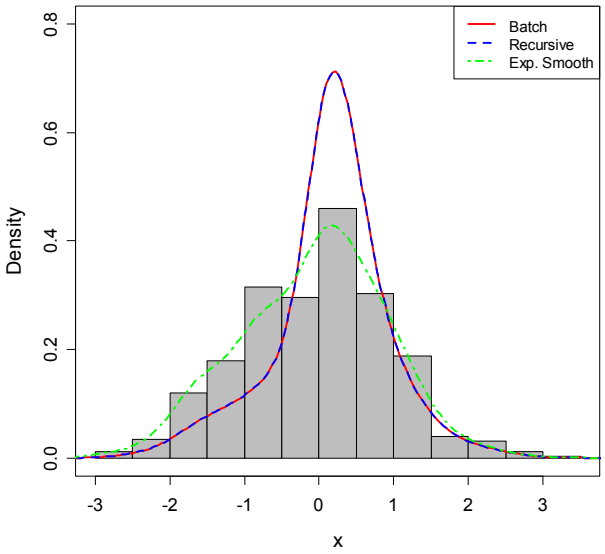

Fig. 3-10 Estimated density function of 500 points drawn from a N(0,1).

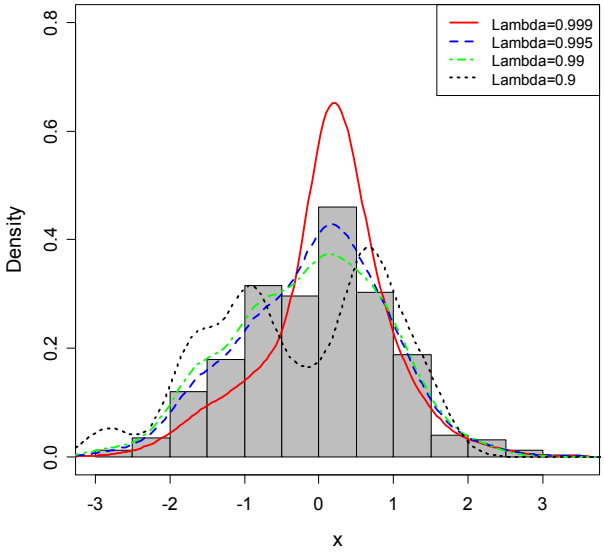

Fig. 3-11 Estimated density function of 500 points drawn from a N(0,1) obtained with different values for λ.

3.3.6.1 Time-adaptive Nadaraya-Watson Estimator

The Nadaraya-Watson Estimator can be converted to a time-adaptive estimator using (3-29). The estimator becomes

$$\hat{f}(y|X=x)_t = \frac{\lambda \cdot \hat{f}_{t-1}(x,y) + (1-\lambda) \cdot \left[K_{h_x}\left(\frac{x-X_i}{h_x}\right) \cdot K_{h_y}\left(\frac{y-Y_i}{h_y}\right) \right]}{\lambda \cdot \hat{f}_{t-1}(x) + (1-\lambda) \cdot K_{h_x}\left(\frac{x-X_i}{h_i}\right)} \tag{3-31}$$

where $f(y|x=X)_t$ is the knowledge of the model at time instant t, which is updated by using recent values of Y and X.

For the wind power forecast problem, we have the following:

I. $\hat{f}_P\left(p_{t+k}|X=x_{t+k|t}\right)_t$: KDF model with knowledge at time step t; and

II. New values of measured wind generation and NWP data are available (e.g., measured wind power generation in the last 24 hours and corresponding NWP data for the same period). This recent data is used to update the knowledge of the model, and the model in (I) becomes $\hat{f}_P\left(p_{t+k}|X=x_{t+k|t}\right)_{t-1}$.

This process is repeated in an online mode (when new values are available).

3.3.6.2 Time-adaptive Quantile-Copula Estimator

The Quantile-Copula estimator can be converted to a time-adaptive estimator using (3-9). The idea is analogous to the one described for the Nadaraya-Watson; however, there is one important aspect related to this CKDE estimator. The quantile transform function — the empirical cumulative distribution function of (3-14) — is transformed to being time-adaptive by using the following equation:

$$F^e(x)_t = \lambda \cdot F^e(x)_{t-1} + (1-\lambda) \cdot \mathrm{I}(x_i \leq x) \tag{3-32}$$

The estimator becomes

$$\hat{f}(y|x=X)_t = \hat{f}_t(y) \cdot \hat{c}_t(u,v) \tag{3-33}$$

where

$$\hat{f}_t(y) = \lambda \cdot \hat{f}_{t-1}(y) + (1-\lambda) \cdot K_h\left(\frac{y-Y_i}{h}\right) \tag{3-34}$$

and

$$\hat{c}_t(u,v) = \lambda \cdot \hat{c}_{t-1}(u,v) + (1-\lambda) \cdot \left[K_1\left(\frac{F_X^e(u)-F_X^e(U_i)}{h_q}\right) \cdot K_2\left(\frac{F_X^e(v)-F_X^e(V_i)}{h_q}\right)\right] \tag{3-35}$$

where $f(y|x=X)_t$ represents the knowledge of the model at time instant *t*, which is updated by using recent values of Y and X.

Note that different values of λ should be defined for (3-32) and (3-34), because the quantile transform of the data should change with a lower rate: otherwise, the model could become unstable. Note that this quantile transform theoretically could create some problems for the time-adaptive model because this result implies that the transforming data structure is also changing. The tests in the subsequent sections will allow a better understanding of this method.

For the wind power forecast problem, we have the following:

$\hat{f}_P\left(p_{t+k}|X = x_{t+k|t}\right)_t$: KDF model with knowledge at time step *t*;

new values of measured wind generation and NWP data are available (e.g., measured wind power generation in the last 24 hours and corresponding NWP data for the same period). These recent data are used to update the knowledge of the model, and the model in (I) becomes $\hat{f}_P\left(p_{t+k}|X = x_{t+k|t}\right)_{t-1}$.

This process is repeated in an online mode (when new values are available).

3.4 Case Studies

3.4.1 Evaluation Framework

3.4.1.1 Benchmark Algorithms

The results obtained with the Nadaraya-Watson estimator and the Quantile-copula estimator will be compared with the linear quantile regression model and the spline quantile regression [27]. The spline quantile regression is a model from the state-of-the-art in wind power forecasting, which consists of a linear quantile regression with the base functions formulated as cubic B-splines, in order to obtain the quantile with proportion α of the forecast errors. Each quantile is modeled as a sum of the nonlinear smooth functions of the forecasted wind power generation (or NWP forecasts). Spline bases are used to approximate each of the smooth functions as a linear combination of base functions.

3.4.1.2 Evaluation Metrics

A framework to evaluate wind power probabilistic forecasts is detailed in [53][54]. The evaluation set consists of a series of quantile forecasts for unique or varying nominal proportions and observations (measured values). The presented classification can be unconditional; however, because several variables might influence the quality of probabilistic forecasts, the evaluation can also become conditional in order to reveal the influence of such variables (e.g., by a look-ahead time step).

Calibration

A requirement for probabilistic forecasts is that the nominal probabilities — or nominal proportions of quantile forecasts — are indeed respected in practice. Obviously, this requirement cannot be assessed on a single evaluation; thus, an evaluation set should be of a significant size. Forecasted probabilities should asymptotically approach the observed probabilities. In other words, in an infinite series of interval forecasts, empirical coverage should equal the pre-assigned probability exactly. This property is commonly referred to as reliability or calibration.

In statistics, the difference between empirical and nominal probabilities is considered to be the bias of the probabilistic forecasting method. Therefore, being unbiased, calibration is translated to the probability forecasts. Bias values are usually calculated for each quantile nominal proportion. However, care must be taken when evaluating calibration: it is not advisable to average the bias over the quantiles. Quantiles below 50% might, for instance, lead to an overestimation, and quantiles above 50% might lead to an underestimation, while the average bias of such prediction would be close to 0.

In order to evaluate quantile forecasts, it is necessary to define the indicator variable. An indicator variable for a quantile forecast $\hat{q}^{\alpha}_{t+k|t}$ is:

$$\xi^{\alpha}_{i,k} = \begin{cases} 1 \; if \; p_{t+k} \leq \hat{q}^{\alpha}_{t+k|t} \\ 0 \; otherwise \end{cases} \tag{3-36}$$

The indicator variable refers to the actual outcome of p_{t+k} at time $t+k$ — that is, whether the quantile covers the actual outcome ("hit") or not ("miss").

Furthermore, these indicators are defined as follows:

$$n_{k,1}^{\alpha} = \#\{\xi_{i,k}^{\alpha} = 1\} = \sum_{i=1}^{N} \xi_{i,k}^{\alpha} \tag{3-37}$$

$$n_{k,0}^{\alpha} = \#\{\xi_{i,k}^{\alpha} = 0\} = N - n_{k,1}^{\alpha} \tag{3-38}$$

— that is, as sums of hits and misses, respectively, for a given horizon *k* over *N* realizations.

A common way of checking calibration is to compare the empirical to the nominal coverage by using the indicators mentioned above, that is:

$$\hat{\alpha}_k^{\alpha} \equiv E[\xi_{i,k}^{\alpha}] = \frac{1}{N}\sum_{i=1}^{N} \xi_{i,k}^{\alpha} \tag{3-39}$$

This way, the estimation $\hat{\alpha}_k^{\alpha}$ of the actual coverage α_k^{α} for a given horizon *k* is obtained using the test set of *N* realizations. The calibration can also be averaged over the entire forecast time horizon. This approach is often used to create calibration diagrams. Calibration diagrams allow the calibration of several quantiles to be summarized, at the same time, providing an overview of whether a particular method systematically underestimates or overestimates uncertainty.

The deviation from the "perfect calibration" (where empirical proportions match nominal or forecasted proportions) or the bias is given by:

$$b_k^{\alpha} = \alpha - \hat{\alpha}_k^{\alpha} \tag{3-40}$$

Sharpness

Sharpness is the tendency of probability forecasts to move toward becoming discrete forecasts, as measured by the mean size of the forecast intervals (distance between quantiles). Quantiles are gathered by pairs in order to obtain intervals with different nominal coverage rates. This step yields an indication of their level of usefulness, where narrow intervals are desired. This measure does not depend on observations. Let $\delta_{t+k|t}^{\alpha} = \hat{q}_{t+k|t}^{1-\alpha/2} - \hat{q}_{t+k|t}^{\alpha/2}$ be the size of the central interval forecast with nominal coverage rate *1-α* estimated at time *t* for lead time *t+k*. A measure of sharpness could then be provided as an average size of intervals:

$$\bar{\delta}_k^{\alpha} = \frac{1}{N}\sum_{i=1}^{N} \delta_{i,k}^{\alpha} \tag{3-41}$$

By having a set of quantile forecasts in pairs, it is possible to summarize sharpness with diagrams, with $\bar{\delta}_k^{\alpha}$ being the function of nominal interval size.

Resolution

The resolution is the concept that evaluates the ability to provide situation-dependent assessment of the uncertainty. However, according to Pinson *et al.* [54], it is not possible to verify this property; therefore, resolution represents the variation of the size of the intervals. The standard

deviation of the interval size for a given look-ahead time step k and coverage rate (1-α) is computed as

$$\sigma_k^{\alpha} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(\delta_{i,k}^{\alpha} - \bar{\delta}_k^{\alpha}\right)^2} \tag{3-42}$$

In this report the standard deviation was computed for each coverage rate.

Skill Score

The objective of scoring rules is to provide the whole information about a model's performance in a single measure [53][54]. A scoring rule for measuring the performance associates a single numerical value $S_c(\hat{f}, p)$ to a predictive distribution $\hat{f}$ if the event p materializes. It can be defined as

$$S_c(\hat{f}', \hat{f}) = \int S_c(\hat{f}'(p), p)\hat{f}(p)dp \tag{3-43}$$

which is the score under $\hat{f}$ when the predictive distribution is $\hat{f}'$.

For nonparametric distributions, in this case represented by a set of m quantiles, Gneiting and Raftery [55] showed that a scoring rules of the form

$$S_c(\hat{f}, p) = \sum_{i=1}^{m}\left(\alpha_i s_i(\hat{q}^{\alpha_i}) + \left(s_i(p) - s_i(\hat{q}^{\alpha_i})\right)\xi^{\alpha_i} + h(p)\right) \tag{3-44}$$

where ξ^{α_i} is the indicator variable for the quantile with proportion α_i, s_i is a nondecreasing function and h arbitrary, which is proper to evaluating this set of quantiles. The score of (3-44) is a positively rewarding score: a higher score value stands for a higher skill.

The skill score used by several authors in the literature (e.g., [17][53][54]) and derived from (3-44) to evaluate wind power quantile forecasts is given by:

$$S_c(\hat{f}_{t+k}, p_{t+k}) = \sum_{i=1}^{m}(\xi^{\alpha_i} - \alpha_i)\left(p_{t+k} - \hat{q}_{t+k}^{\alpha_i}\right) \tag{3-45}$$

where $s_i(p) = p_{t+k}$ and $h(p) = -\alpha p_{t+k}$. The higher the scoring rule, the better, and the maximum value is 0 for perfect probabilistic forecasts.

The skill score can be computed for each look-ahead time step by using with the following:

$$S_{c_k} = \frac{1}{N}\sum_{t=1}^{N} S_c(\hat{f}_{t+k}, p_{t+k}) \tag{3-46}$$

where N is the number of samples from the evaluation set.

Pinson *et al*. [54] mentioned that using a unique proper skill score allows one to compare the overall skill of competitive approaches, given that scoring rules encompass all the aspects of probabilistic forecast evaluation. However, it does not inform on the contributions of calibration or sharpness and resolution to the skill score. Hence, these authors suggested that calibration should be assessed in a first analysis (as the primary requirement), and then the information provided by the skill score allows users to derive conclusions about the remaining metrics.

3.4.2 Evaluation Results: NREL's EWITS Study

3.4.2.1 Case-Study Description

The wind power data used are day-ahead wind power point forecasts and realized wind power generation for 15 sites in the state of Illinois within the Midwest Independent System Operator's footprint for 2006 as obtained from NREL's EWITS study [56]. The data were produced by combining a mesoscale NWP model with a composite power curve for a number of potential sites for wind power farms. The day-ahead forecasts were generated based on observed forecast errors from four real wind power plants. The resulting Markov chain forecast models for each of the four sites were randomly assigned to the sites in the dataset. The data methodology is explained in [57].

We use the wind power data (forecasts and realized generation) for the period from January to August to train the uncertainty forecast models. The months between September and December are used as a test dataset. The main characteristics of the two datasets are presented in Table 3-1. The explanatory variable in this case study is the point forecast.

Table 3-1 Statistical characteristics of the NREL training and testing dataset.

Variables	Nº Points	Mean (p.u.)	Median (p.u.)	Std. Dev. (p.u.)	Skewness	Kurtosis	IQR[a] (p.u.)
Train Dataset							
Forecast	5088	0.327	0.282	0.176	0.847	3.077	0.250
Realized	5088	0.325	0.256	0.256	0.710	2.380	0.393
Test Dataset							
Forecast	3672	0.333	0.303	0.185	0.631	2.679	0.274
Realized	3672	0.337	0.280	0.250	0.568	2.196	0.397

[a] IQR = Inter-quantile range; p.u. = power unit.

3.4.2.2 Offline Evaluation Results

For reasons of comparison, the probabilistic forecast is represented through a set of quantiles ranging from 5% to 95% with a 5% increment, as depicted in Fig. 3-12 in the form of a set of interval forecasts.

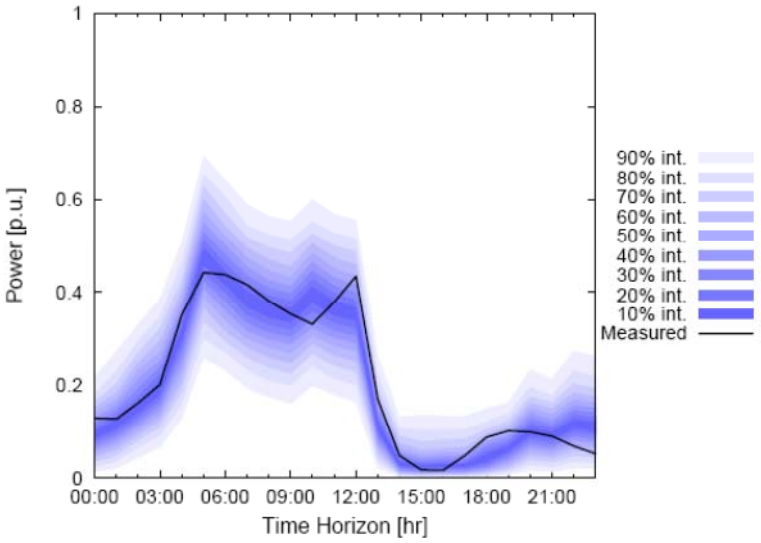

Fig. 3-12 Probabilistic forecast for NREL dataset obtained with the NW estimator.

The Kernel function used in the Nadaraya-Watson (NW) and Quantile-Copula estimators were Chen's beta kernels from (3-17) for both realized and forecasted wind power. The kernel size was 0.001 for both variables (determined experimentally by trial and error). The tests performed with different bandwidths showed that by changing the kernel bandwidth, the model changes from underestimation to overestimation and vice-versa.

Fig. 3-13 depicts the calibration diagram averaged for the whole time horizon (24 hours) for the probabilistic forecasts obtained with the linear quantile regression (Linear QR), splines quantile regression (Splines QR), NW estimator (NW), and Quantile-copula (QC) estimator. Note that what is depicted in the diagram is the difference from the "perfect calibration" (according to [3-40]).

All of the models presented in Fig. 3-13 show a deviation from the "perfect calibration" below 5%, which, according to Juban *et al*. [14], is equivalent to what is found in the literature. For the quantiles above 55%, the NW and QC estimators present a lower deviation than the quantile regression methods. For quantiles below the median, the splines QR is competitive with the KDF methods, and for some quantiles it achieves the lowest deviation. For these quantiles, the QC approach also presents a lower deviation than the NW approach. On average, the methods overestimate (nominal proportions greater than empirical) the quantiles.

Fig. 3-14 depicts a sharpness diagram where the x-axis is the nominal coverage of the forecast interval (1-α), and the y-axis is the average size of the intervals. In this case, the desired outcome is to have intervals with a smaller size for all coverage rates. In terms of sharpness, the forecasted quantiles presented relatively narrow amplitudes in all methods, although QR splines presented a lower sharpness. It is important to note that Juban *et al*. [14] found a trade-off between reliability and sharpness, meaning that improving the reliability will generally degrade the sharpness and vice-versa.

Fig. 3-15 depicts the resolution diagram where the x-axis is the nominal coverage of the forecast interval (1-α), and the y-axis is the standard deviation of the intervals. The linear QR presented

the lowest standard deviation and, consequently, the lowest resolution. The other methods presented almost the same resolution, with a slight advantage for the splines QR.

The diagrams depicted in Figs. 3-13 to 3-15 are averaged over the forecast time horizon. However, the uncertainty of the wind power forecast is influenced by several factors, such as the look-ahead time step. Therefore, Figs. 3-16 to 3-18 depict calibration, sharpness, and resolution for look-ahead time step *t+17h*. Note that the calibration diagram of Fig. 3-16 is different: the y-axis is the empirical proportion computed with (3-39). The "ideal" line is the perfect match between nominal and empirical proportions. Calibration is indicated by the proximity of the plotted curve to the "ideal" diagonal. If the curve lies below the line, over-forecasting is indicated (i.e., forecasted quantiles are too high); points above the line indicate under-forecasting (i.e., forecasted quantiles are too low).

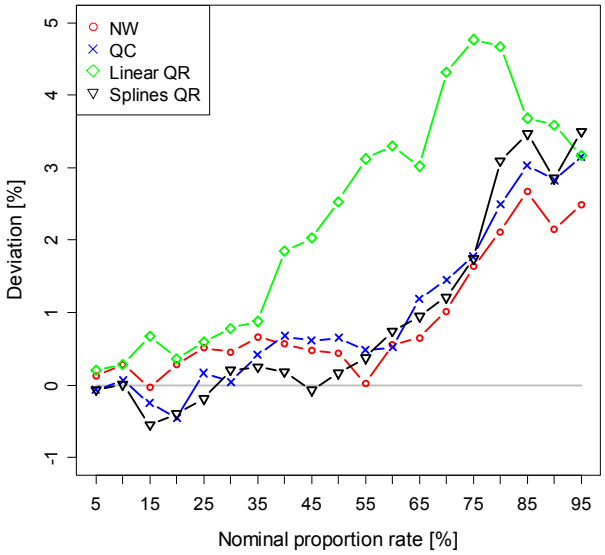

Fig. 3-13 Calibration diagram for the offline test with NREL data.

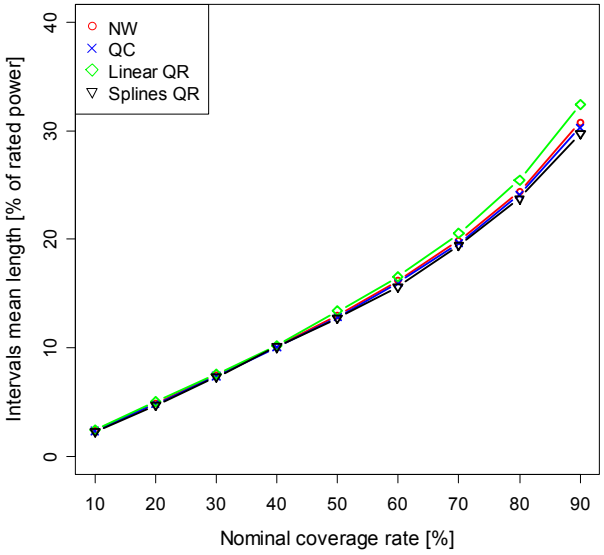

Fig. 3-14 Sharpness diagram for the offline test with NREL data.

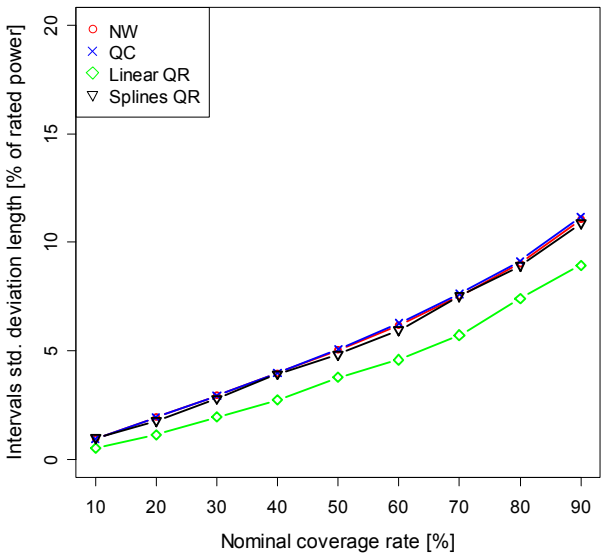

Fig. 3-15 Resolution diagram for the offline test with NREL data.

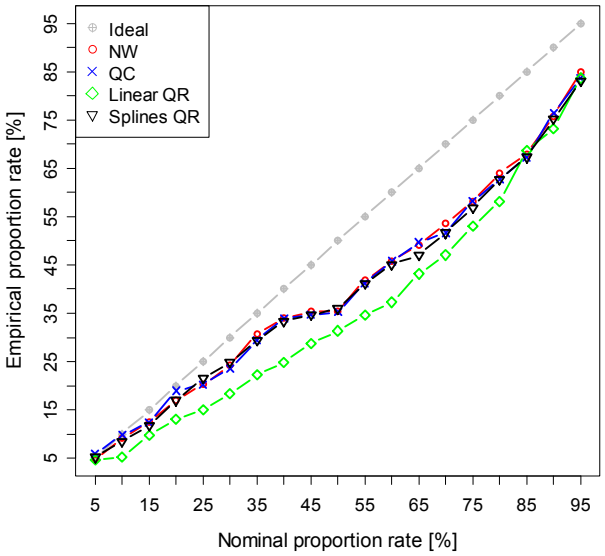

Fig. 3-16 Calibration diagram for look-ahead time step t+17h.

As shown in Fig. 3-16, when computed for a specific look-ahead time step, calibration presents a higher deviation from the "ideal" diagonal when compared to the calibration computed over the whole time horizon (depicted in Fig. 3-13). This figure depicts a situation with quantiles overestimation (nominal proportion higher than empirical); however, in some hours, there are also underestimations.

The difference depicted in Fig. 3-17 between the methods in terms of sharpness is more pronounced. The same was verified for the resolution in Fig. 3-18. The calibration, sharpness, and resolution for other look-ahead time steps *t+6h* and *t+22h* are presented in Appendix A.

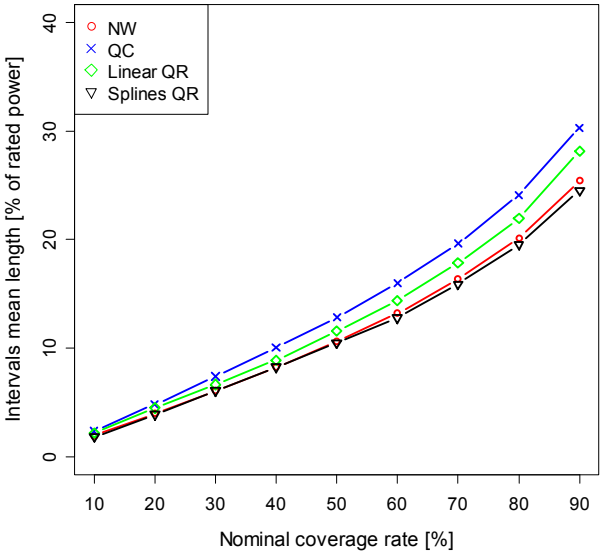

Fig. 3-17 Sharpness diagram for look-ahead time step t+17h.

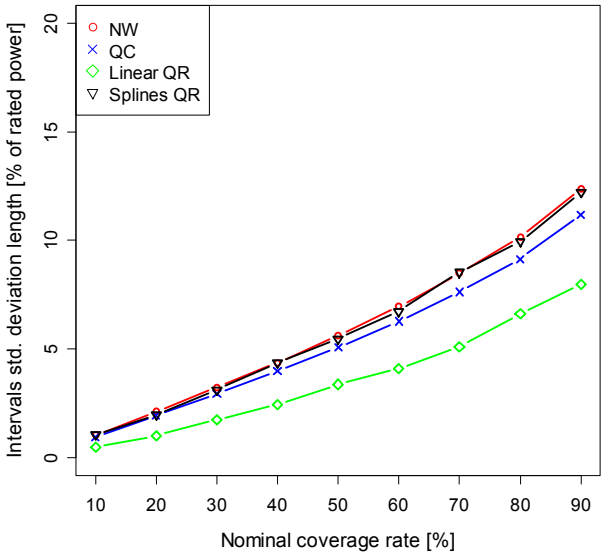

Fig. 3-18 Resolution diagram for look-ahead time step t+17h.

3.4.2.3 Time-adaptive Test: Proof of Concept

The aim of this sub-section is to demonstrate that the time-adaptive concept works in conditional KDE and can be applied to the wind power problem. The same data described in Table 3-1 was used for this test. However, in order to introduce a change artificially in the data structure, we "disconnected" two sites (one of 211.6 MW and another of 616.1 MW, in a 5.19-GW total) during January–September, and these two sites were only "connected" after October.

This situation was created artificially; however, it reproduces a situation that could happen to a system operator. For instance: a system operator is receiving forecasts from 13 wind farms; then, these forecasts are summed up and estimates of the uncertainty are associated to the total wind power generation; then, in October two wind farms are connected to the grid, and in this case, the knowledge from past observations is no longer valid. By using a time-adaptive model, the system operator is able to adapt to the new situation without requiring an offline training of the model. Moreover, the system operator will need to wait several months in order to have sufficient data to perform the offline training.

The results will only be analyzed in terms of calibration, given that the major impact of this situation is in an underestimation and overestimation of the quantiles. Fig. 3-19 depicts the calibration diagram of the probabilistic forecasts obtained with the offline NW estimator and also with the time-adaptive NW estimator with three different values of λ. Fig. 3-20 depicts the calibration diagram for the offline and time-adaptive QC estimator. The preliminary tests showed that the value of λ for the empirical cumulative distribution function should be very low; in this case, a value equal to 0.9995 was used.

Due to the increase in the wind power generation because of the connection of two wind farms, it is expected that the offline approach gives an underestimation of the quantiles for values below the 50% quantile and an overestimation of the quantiles for greater values. As an example, the 95% quantile means that the probability of having wind generation above its value is only 5%; however, the empirical quantiles estimated with the offline approach say that this probability is 13%. This finding means that the probability of having more wind generation in the system is higher than what is predicted. The opposite situation is verified for the quantiles below 50%, for example, the 10% quantile means that with 90% probability, the wind generation will be above its value; however, the empirical proportion for the offline approach says that this probability is 15.1%.

The time-adaptive approach can incorporate the recent information and discount the old information (controlled with λ). Therefore, underestimations and overestimations are corrected using this approach.

For the NW estimator, the calibration obtained with λ equal to 0.999 and 0.995 is much better than that obtained by the offline approach. For instance, for the quantile 95%, the empirical proportions obtained with the time-adaptive approach is 92.3% with λ=0.999 and 91.2% with λ=0.995. For the 15% quantile, the empirical proportions are 12.7% (λ=0.999) and 13.6% (λ=0.995). The results for the QC estimator are different in terms of calibration; in this case, the estimator presents better calibration performance for λ equal to 0.999.

As a first consideration, the use of a lower value for λ would be recommended in order to foster quick learning of the new data structure. However, as the results for 0.99 show, this lower value leads to results that are comparable to the offline approach. The reason is that the KDF becomes numerically unstable, and thus it is unable to assimilate the recent information properly. The main conclusion is that a value of λ near 1 should be used and, in the case of concept change, this value could be reduced, but after a while it should be increased again.

In Appendix A, the calibration for look-ahead time steps *t+15h* and *t+20h* are presented.

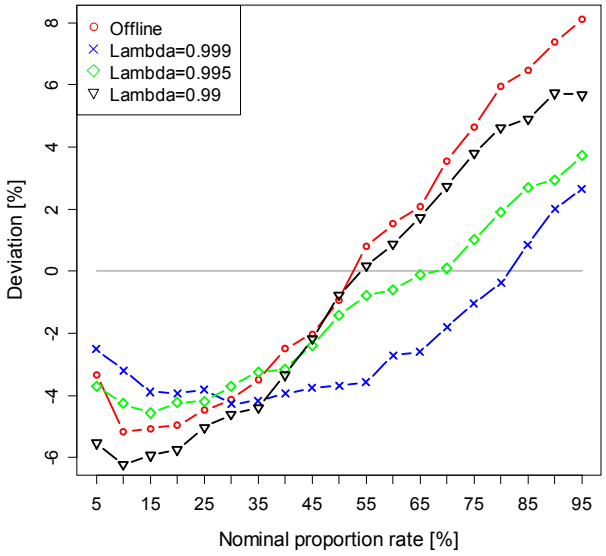

Fig. 3-19 Calibration diagram for the NREL dataset with concept change and NW estimator.

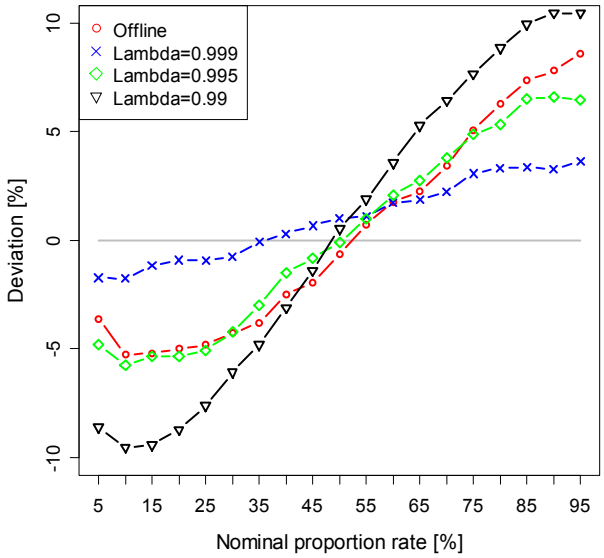

Fig. 3-20 Calibration diagram for the NREL dataset with concept change and QC estimator.

3.4.3 Evaluation Results: Midwest Wind Farm

3.4.3.1 Case-Study Description

The wind power data is from a large-scale wind farm located in a flat terrain in the U.S. Midwest. The wind farm was divided into two "sub-wind farms" named Wind Farm A (WFA) and Wind Farm B (WFB).

The complete dataset (SCADA and NWP) available for this project correspond to the period between January 2, 2009, and February 20, 2010. The NWP data was generated by the WRF model at Argonne National Laboratory and consists of several weather variables (e.g., wind speed, direction, temperature) for 11 geographically distributed NWP points.

In these case studies, the reference NWP point used as input for WFA was point number 8, and for WFB, point 6 was used. The temporal horizon of the predictions used was as follows: the results presented from Section 3.4.3.2 to Section 3.4.3.4 were obtained for the temporal horizon of *t+18h* up to *t+42h*; for the results presented in Sections 3.4.3.5 and 3.4.3.6, the time horizon was *t+6h* up to *t+48h*.

For market purposes, the required temporal resolution of wind power forecasts is usually one hour. Both the SCADA and NWP data we used have temporal resolution of 10 minutes, so a simple average of the 10-minutes data was performed to produce hourly data.

The explanatory variable in this case study, unless otherwise stated, is the wind speed forecast from the NWP model.

3.4.3.2 Evaluation with Different Kernel Types

Comparisons between six different kernel functions were performed for various combinations of the kernel bandwidths, σ, for wind speed and power. These kernel size values were determined experimentally (by trial and error) and using as a starting point the values suggested by the function *cde.bandwidths* from the R package "hdrcde" [58].

The Kernel functions used in the Nadaraya-Watson (NW) and Quantile-Copula (QC) estimators were Chen's gamma and beta kernels (*Chen 1* and *Chen 2* from equations [3-17] [3-21] and [3-18] [3-22], respectively) and Boundary kernel (proposed by Zhang and Karunamuni [43]) for both wind speed and power variables; Chen's beta kernels with X swapped with x (*Beta 1x* and *Beta 2x* from equations [3-17] and [3-20]) and *jh* kernel (developed by Jones and Henderson [44]) for the wind power variable. The kernel size values were determined experimentally (i.e., testing offline).

The most important results using the NW and QC estimators for dataset A (training period from January 2, 2009, until July 31, 2009, and testing period between August 1, 2009, and February 20, 2010) in both wind farms and day-ahead forecasts (from *t+18h* to *t+42h*), are presented below. Results obtained for the other two estimators from the state-of-the-art, Spline Quantile Regression (QR) and Linear Quantile Regression (LQR), were the same among all kernels used for each bandwidth combination.

Kernel Sizes: $(h_{Power;}\ h_{WindSpeed}) = (0.002; 0.04)$

Figs. 3-21 through 3-24 show results of offline testing of calibration at the stated kernel sizes using the WFA and WFB datasets.

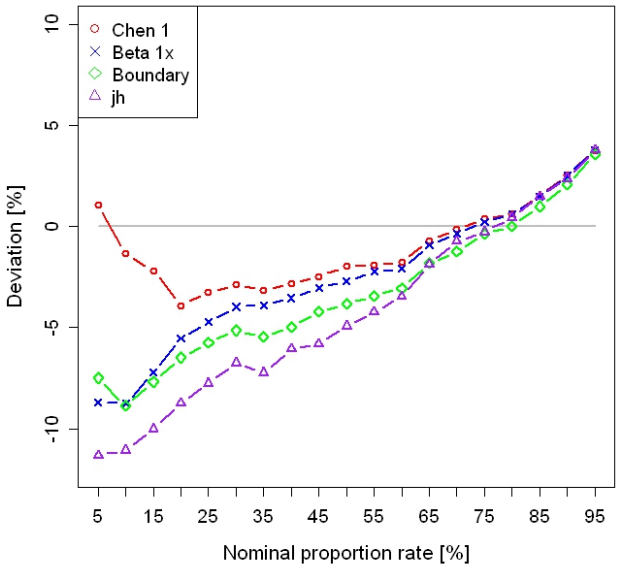

Fig. 3-21 Calibration diagram using NW estimator, for the offline test with WFA dataset A.

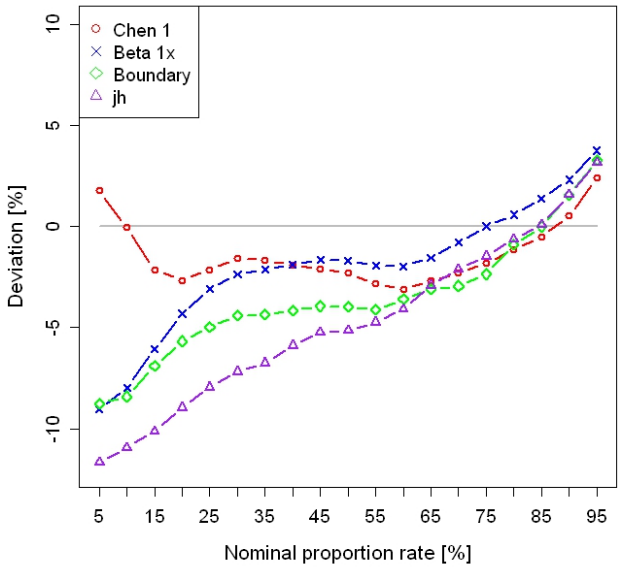

Fig. 3-22 Calibration diagram using QC estimator, for the offline test with WFA dataset A.

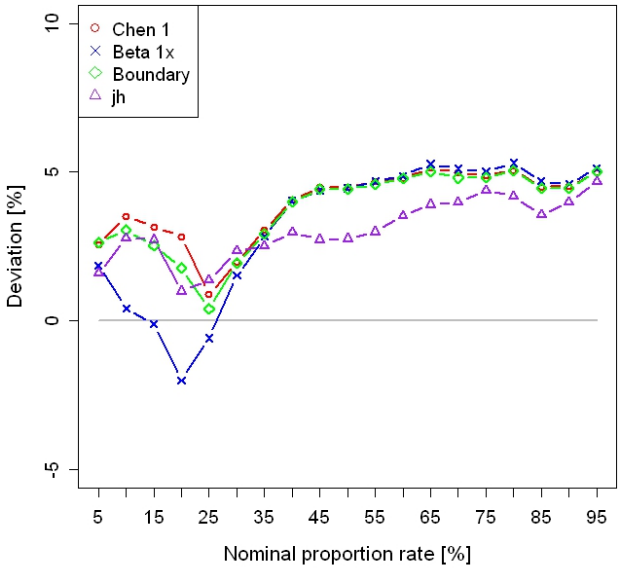

Fig. 3-23 Calibration diagram using NW estimator, for the offline test with WFB dataset A.

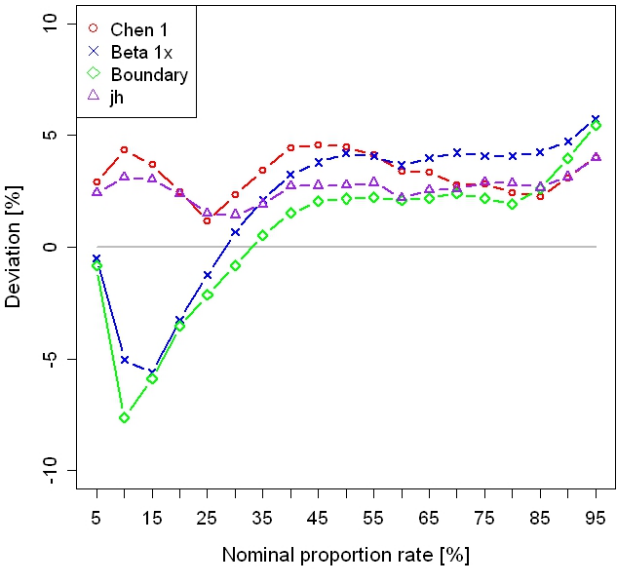

Fig. 3-24 Calibration diagram using QC estimator, for the offline test with WFB dataset A.

For this kernel size choice, only *Chen 1*, *Beta 1x*, *Boundary,* and *jh* are able to perform results. Fig. 3-21 and Fig. 3-22 depict the calibration obtained for WFA using the NW and QC

estimators, respectively. In the former, *Chen 1* performs better than do the other kernels; however, in QC, *Beta 1x* becomes the better performer between quantiles 40%–80%. In both the NW and QC estimators, the *jh* kernel has the worst performance. For this wind farm, there is a tendency to underestimate the quantiles.

As for WFB, Fig. 3-23 shows that *jh* is the kernel with the best calibration using the NW estimator for quantiles above 35%; however, when using QC for quantiles above 30%, it is the *Boundary* kernel instead, as depicted in Fig. 3-24. For this wind farm, there is a tendency to overestimate the quantiles.

Figs. 3-25 through 3-28 show results of offline testing of sharpness using the WFA and WFB datasets.

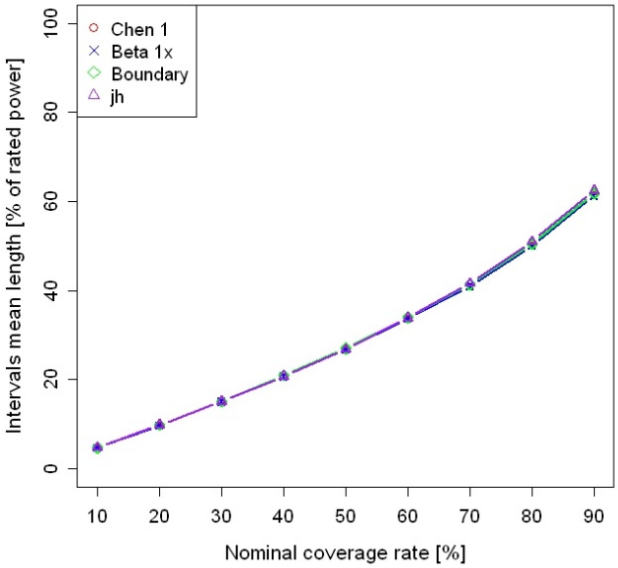

Fig. 3-25 Sharpness diagram using NW estimator, for the offline test with WFA dataset A.

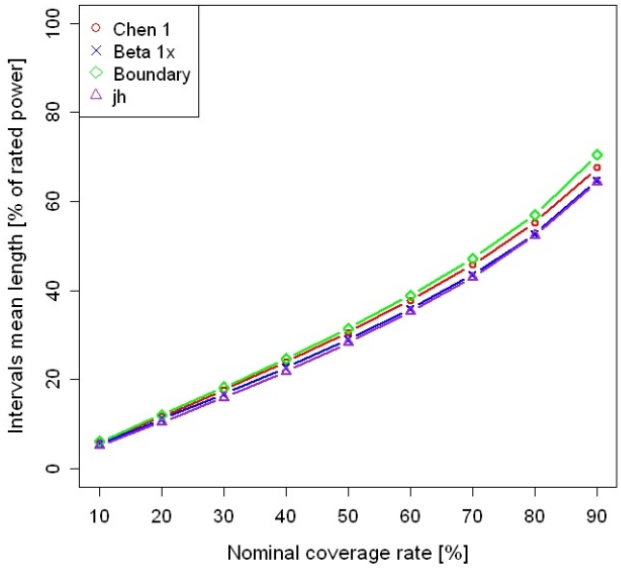

Fig. 3-26 Sharpness diagram using QC estimator, for the offline test with WFA dataset A.

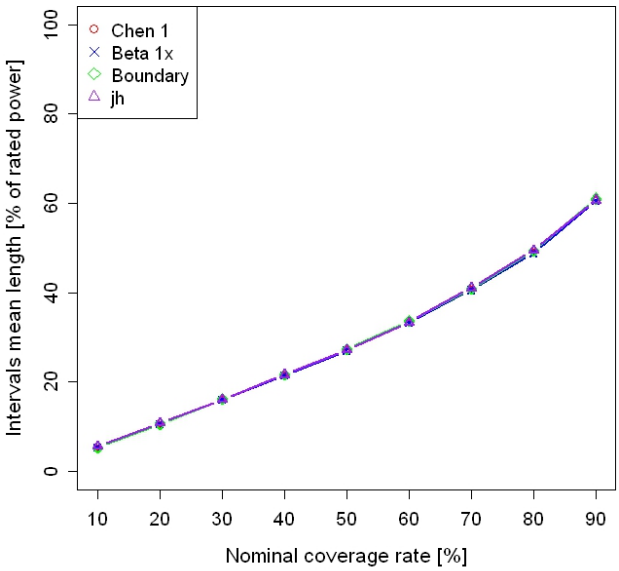

Fig. 3-27 Sharpness diagram using NW estimator, for the offline test with WFB dataset A.

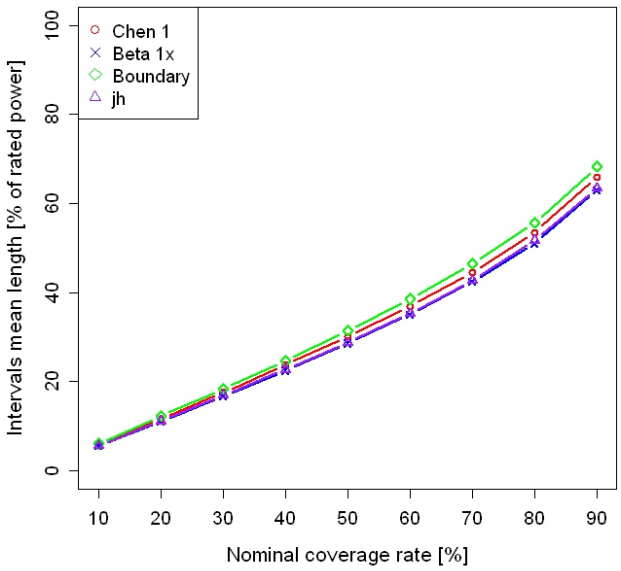

Fig. 3-28 Sharpness diagram using QC estimator, for the offline test with WFB dataset A.

For both wind farms, sharpness results using the NW estimator are the same for all kernels. As illustrated in Fig. 3-26 and Fig. 3-28, *Boundary* performs worse in terms of sharpness in both wind farms when using the QC estimator. Moreover, this kernel has a worse resolution for QC, as depicted in Fig. 3-30 and Fig. 3-32, and when using the NW estimator, all kernels present the same resolution results in both wind farms (Figs. 3-29 through 3-32).

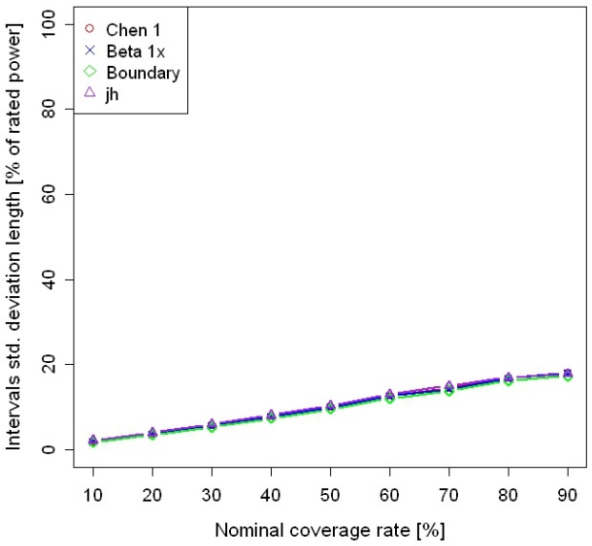

Fig. 3-29 Resolution diagram using NW estimator, for the offline test with WFA dataset A.

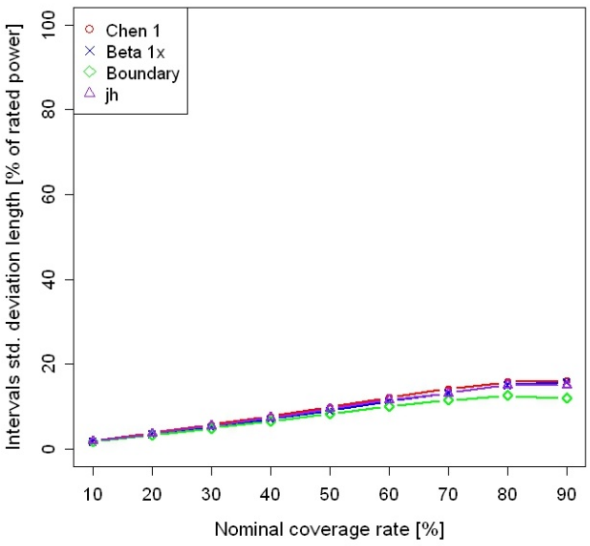

Fig. 3-30 Resolution diagram using QC estimator, for the offline test with WFA dataset A.

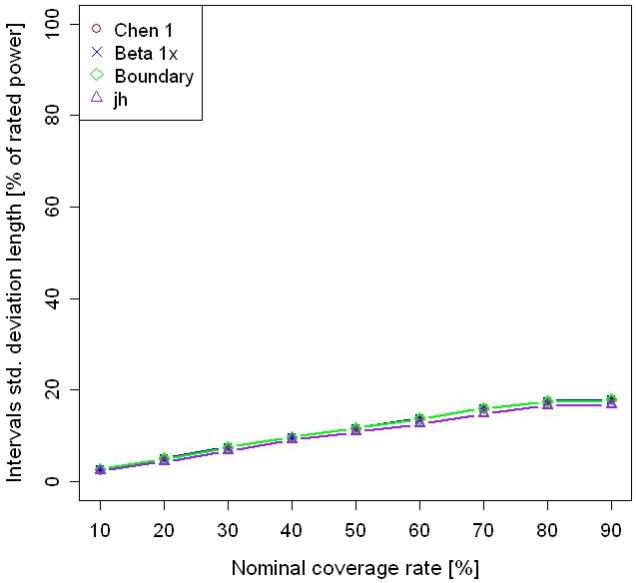

Fig. 3-31 Resolution diagram using NW estimator, for the offline test with WFB dataset A.

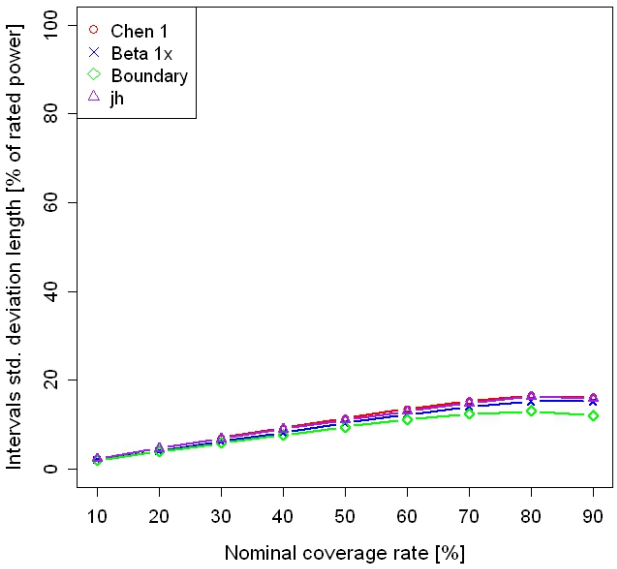

Fig. 3-32 Resolution diagram using QC estimator, for the offline test with WFB dataset A.

In terms of the skill score (Figs. 3-33 through 3-36), the *Boundary* kernel performs the worst when using the QC estimator for both wind farms. Fig. 3-33 shows that the WFA testing for NW presents worse performance in the skill score using the *jh* kernel. *Chen 1* is the best not only for

NW but also for QC. Fig. 3-36, on the other hand, shows that the *jh* kernel has a better skill score for WFB when using QC. However, using NW for WFB, all kernels have a similar skill score.

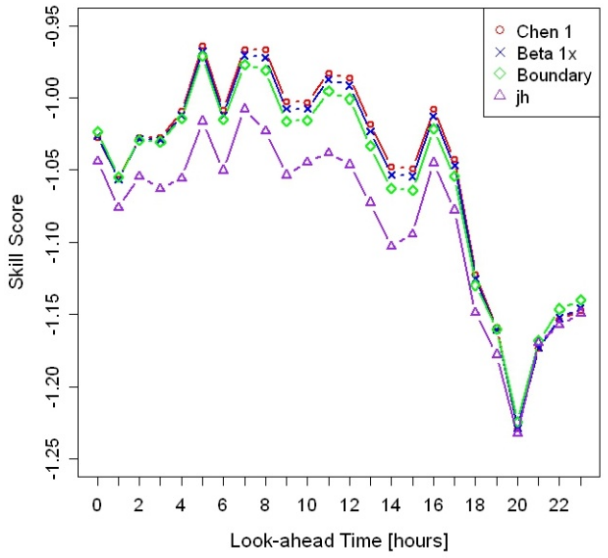

Fig. 3-33 Skill score diagram using NW estimator, for the offline test with WFA dataset A.

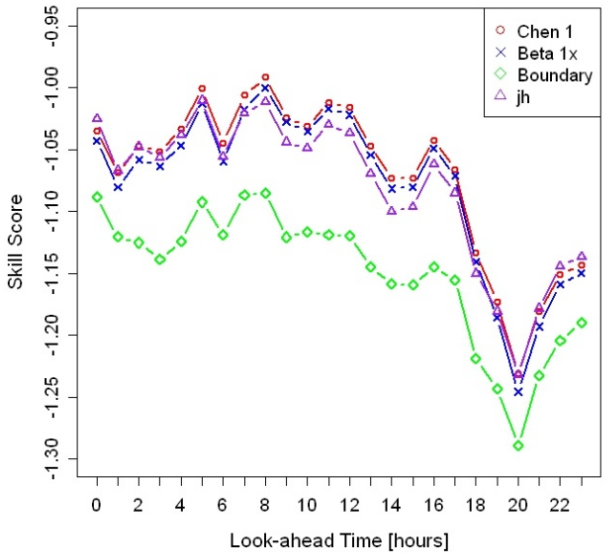

Fig. 3-34 Skill score diagram using QC estimator, for the offline test with WFA dataset A.

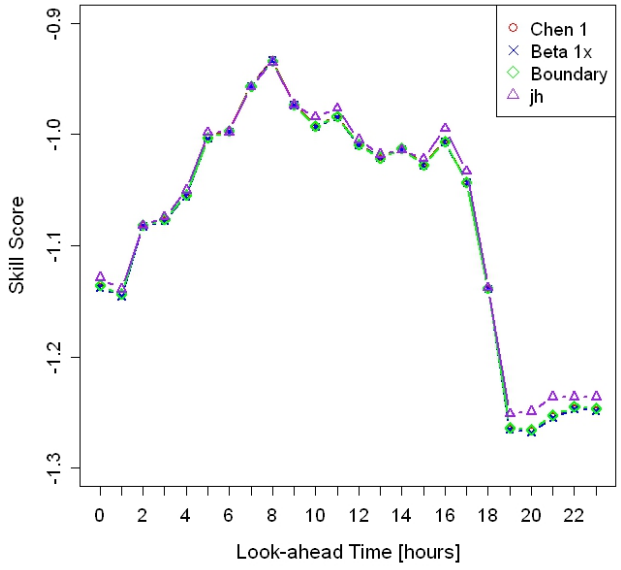

Fig. 3-35 Skill score diagram using NW estimator, for the offline test with WFB dataset A.

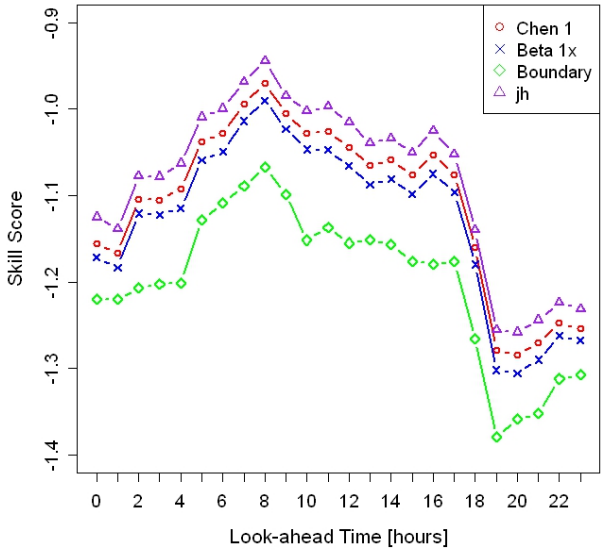

Fig. 3-36 Skill score diagram using QC estimator, for the offline test with WFB dataset A.

For this kernel size, the main conclusions are that for WFA, *Chen 1* has the best overall calibration, and there is a tendency to underestimate the quantiles. In terms of sharpness and

resolution, the *Boundary* kernel performs worse than the others do when using the QC estimator; moreover, using the NW estimator leads to the same sharpness and resolution results between kernels, in both wind farms. As for the skill score, *Boundary* has the worst performance for QC in both wind farms, and in WFA *Chen 1* presents the best performance in both estimators; in addition to this, for WFB using NW skill score results are the same for all kernels.

Kernel Sizes: ($h_{Power;}$ $h_{WindSpeed}$) = (0.004; 0.02)

Figs. 3-37 through 3-40 show results of offline testing of calibration at the stated kernel sizes using the WFA and WFB datasets.

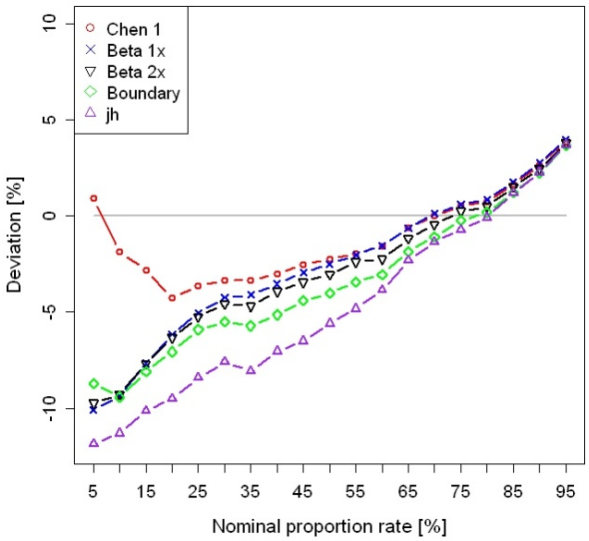

Fig. 3-37 Calibration diagram using NW estimator, for the offline test with WFA dataset A.

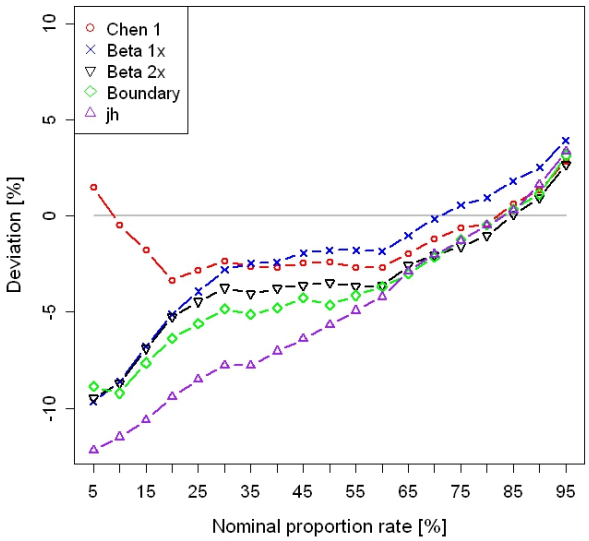

Fig. 3-38 Calibration diagram using QC estimator, for the offline test with WFA dataset A.

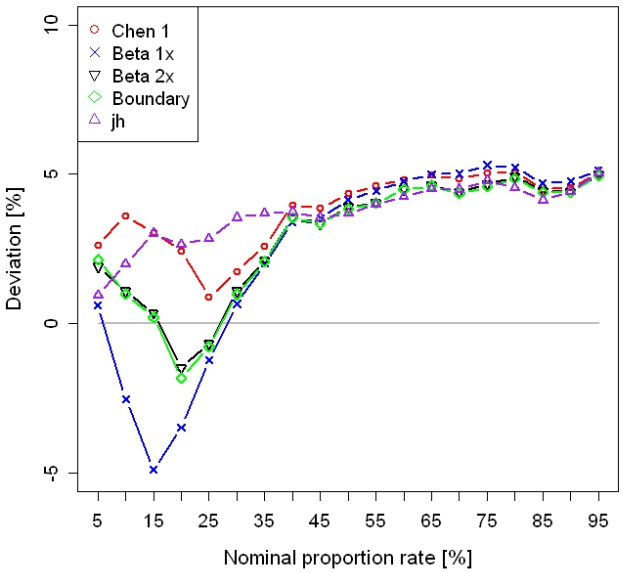

Fig. 3-39 Calibration diagram using NW estimator, for the offline test with WFB dataset A.

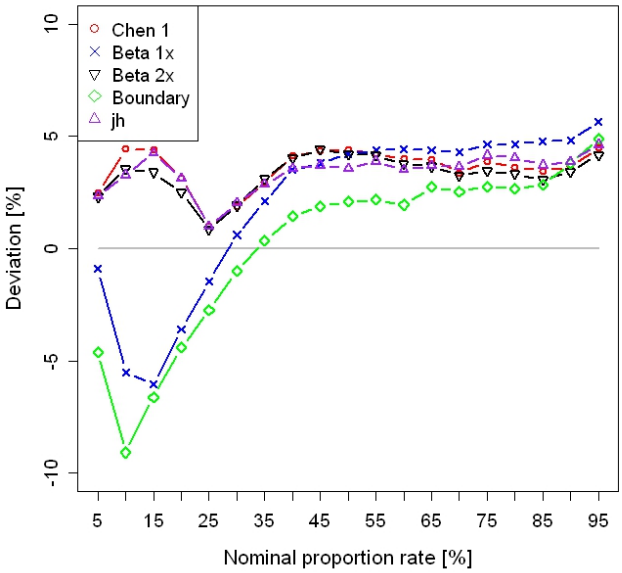

Fig. 3-40 Calibration diagram using QC estimator, for the offline test with WFB dataset A.

For this kernel size choice, only *Chen 2* is not able to deliver results. Fig. 3-37 and Fig. 3-38 depict the calibration obtained for WFA using the NW and QC estimators, respectively. In the former, *Chen 1* performs better than do the other kernels; however, in QC, *Beta 1x* becomes the better performer for quantiles above 30%. In both the NW and QC estimators, the *jh* kernel has the worst performance. For this wind farm, there is a tendency to underestimate the quantiles.

As for WFB, Fig. 3-39 and Fig. 3-40 show that *Boundary* is the kernel with best overall calibration. For this wind farm, there is a tendency to overestimate the quantiles.

Figs. 3-41 through 3-44 present the sharpness results for WFA and WFB.

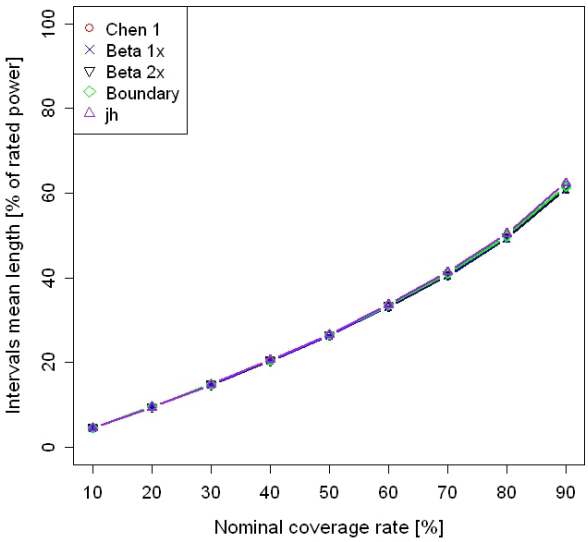

Fig. 3-41 Sharpness diagram using NW estimator, for the offline test with WFA dataset A.

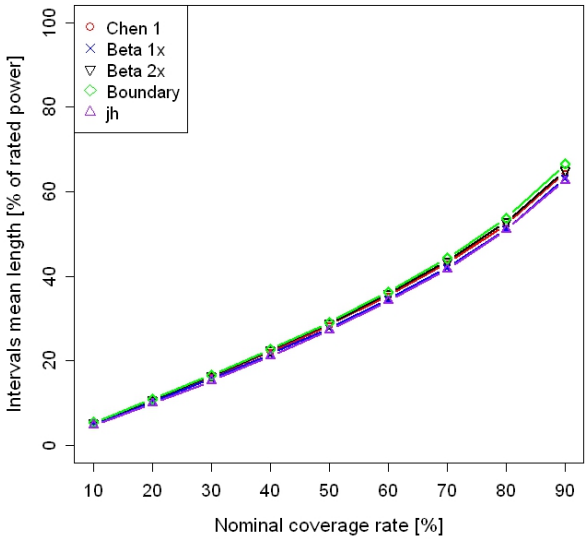

Fig. 3-42 Sharpness diagram using QC estimator, for the offline test with WFA dataset A.

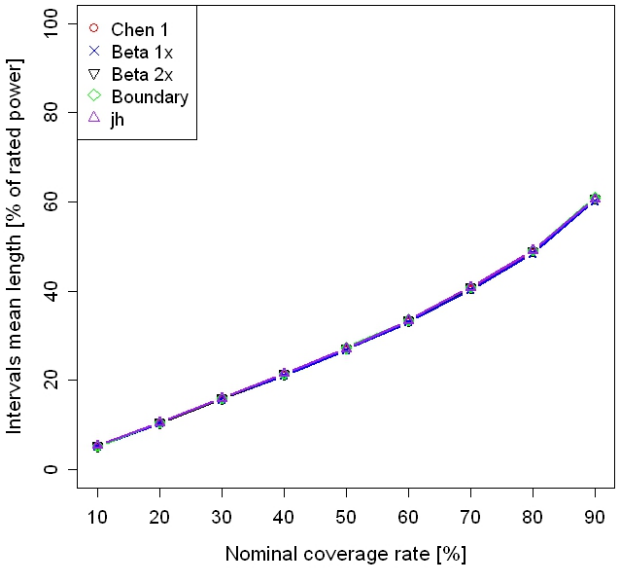

Fig. 3-43 Sharpness diagram using NW estimator, for the offline test with WFB dataset A.

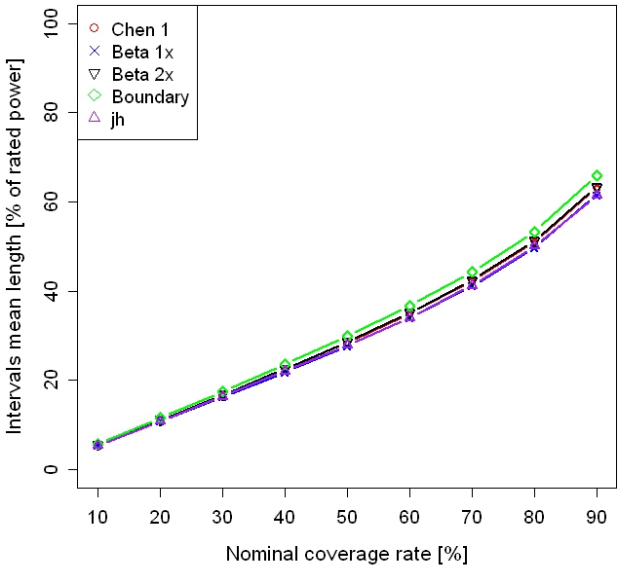

Fig. 3-44 Sharpness diagram using QC estimator, for the offline test with WFB dataset A.

For both wind farms, the sharpness and resolution results using the NW estimator are the same for all kernels. As illustrated in Fig. 3-42 and Fig. 3-44, *Boundary* performs worse in terms of

sharpness in both wind farms when using the QC estimator and, as depicted in Fig. 3-46 and Fig. 3-48, this kernel has the worst resolution for QC, as well.

Figs. 3-45 through 3-48 present the resolution results.

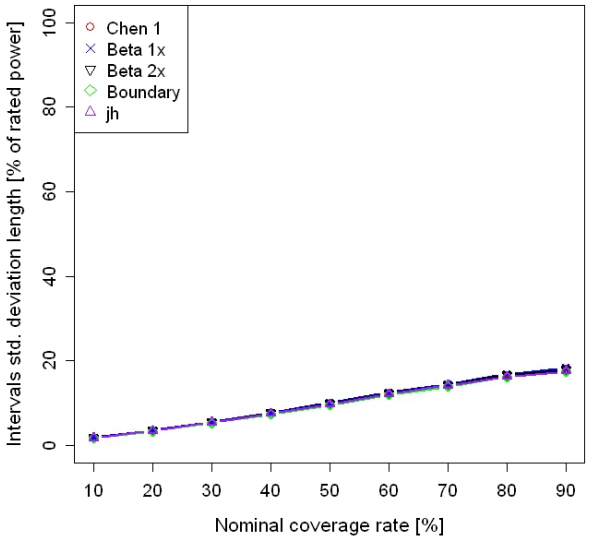

Fig. 3-45 Resolution diagram using NW estimator, for the offline test with WFA dataset A.

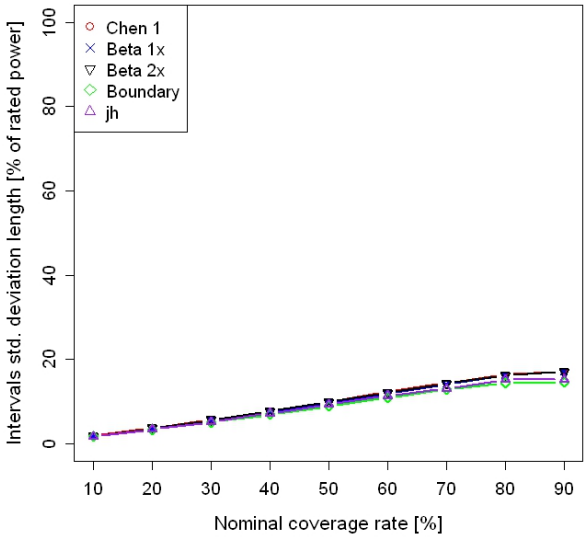

Fig. 3-46 Resolution diagram using QC estimator, for the offline test with WFA dataset A.

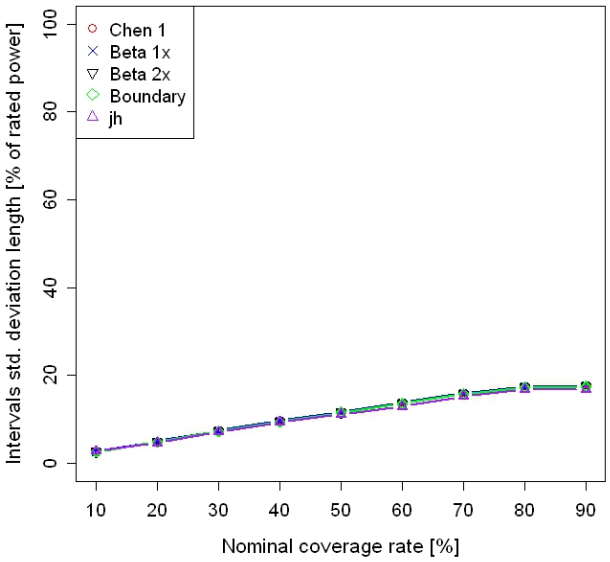

Fig. 3-47 Resolution diagram using NW estimator, for the offline test with WFB dataset A.

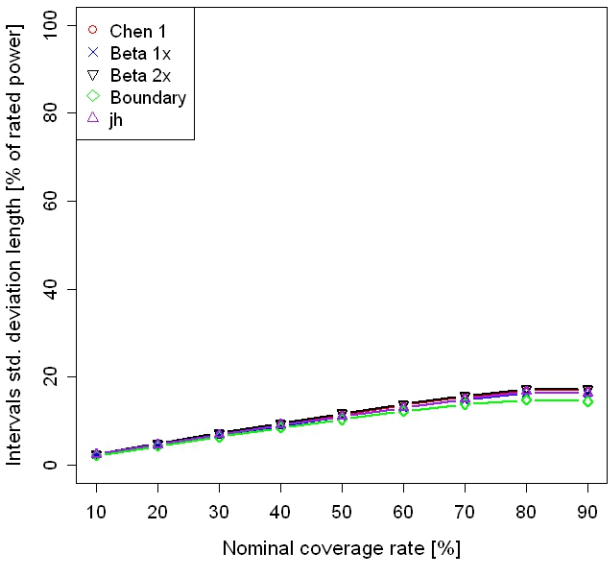

Fig. 3-48 Resolution diagram using QC estimator, for the offline test with WFB dataset A.

In terms of the skill score, the *Boundary* kernel is the worst when using the QC estimator for both wind farms. Fig. 3-49 shows that for WFA using NW, the *jh* kernel presents the worst skill score performance. For this wind farm, *Chen 1* is the best not only for NW but also for QC. Fig. 3-52, on the other hand, shows that the *jh* kernel has a slightly better skill score for WFB when using QC. However, when using NW for WFB, all kernels have a similar skill score (Figs. 3-49 through 3-52).

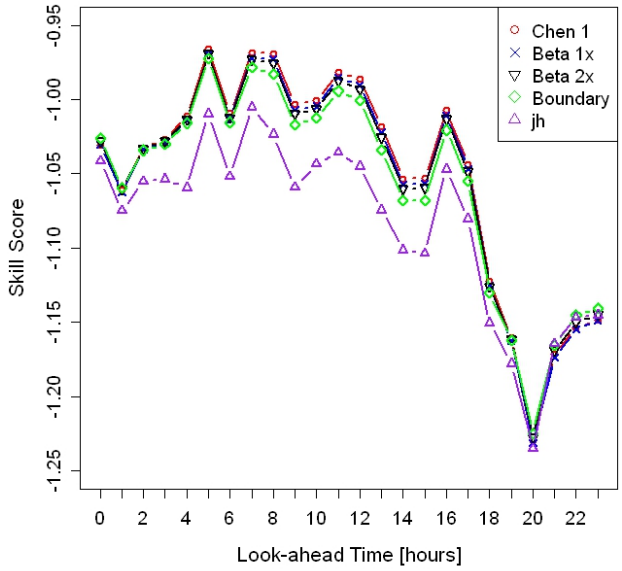

Fig. 3-49 Skill score diagram using NW estimator, for the offline test with WFA dataset A.

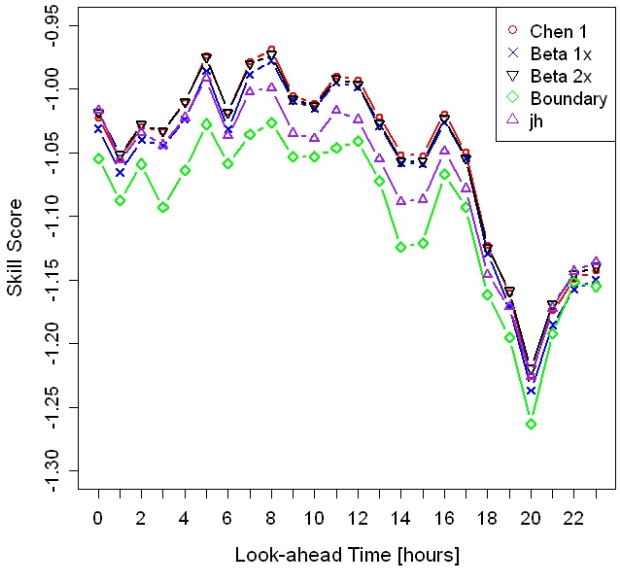

Fig. 3-50 Skill score diagram using QC estimator, for the offline test with WFA dataset A.

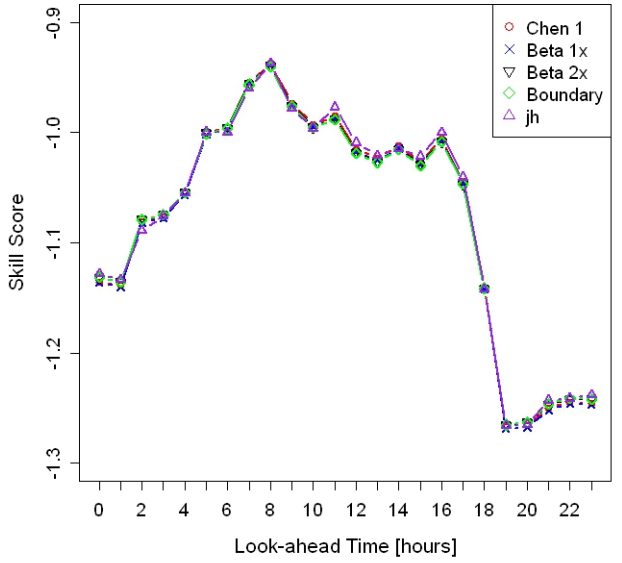

Fig. 3-51 Skill score diagram using NW estimator, for the offline test with WFB dataset A.

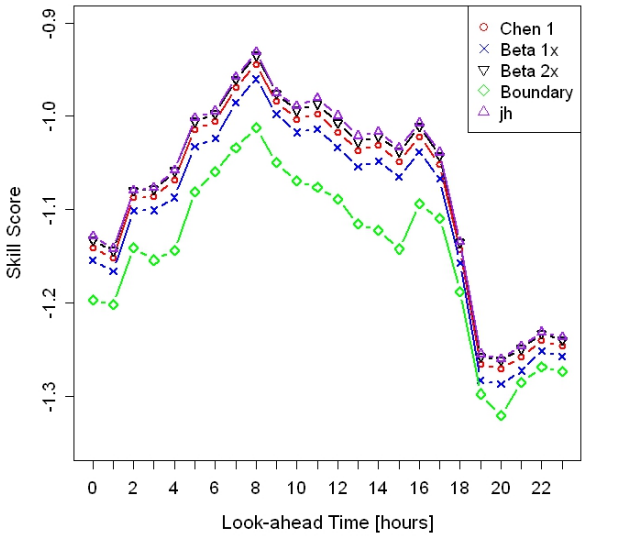

Fig. 3-52 Skill score diagram using QC estimator, for the offline test with WFB dataset A.

The main conclusions for this kernel size choice are the same as in the previous case. Therefore, for WFA, *Chen 1* has the best overall calibration, and there is a tendency to underestimate the quantiles; for WFB, the kernel with better calibration performance is *Boundary*, and there is an overestimation of the quantiles. In terms of sharpness and resolution, the *Boundary* kernel

performs worse than the others do when using the QC estimator; moreover, using the NW estimator leads to the same sharpness and resolution results between kernels in both wind farms. As for the skill score, *Boundary* has the worst performance for QC in both wind farms, and in WFA, *Chen 1* presents better performance in both estimators. In addition to this result, for WFB, when using the NW skill score, results are the same for all kernels.

Kernel Sizes: (h_{Power}; $h_{WindSpeed}$) = (0.004; 1)

Figs. 3-53 through 3-56 show results of offline testing of calibration at the stated kernel sizes using the WFA and WFB datasets.

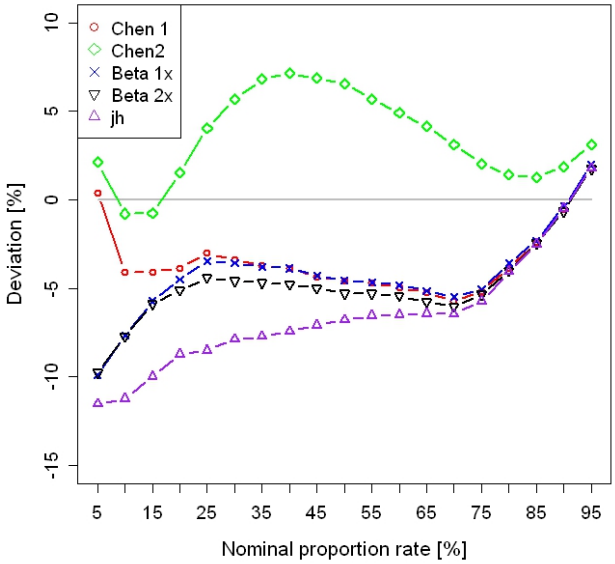

Fig. 3-53 Calibration diagram using NW estimator, for the offline test with WFA dataset A.

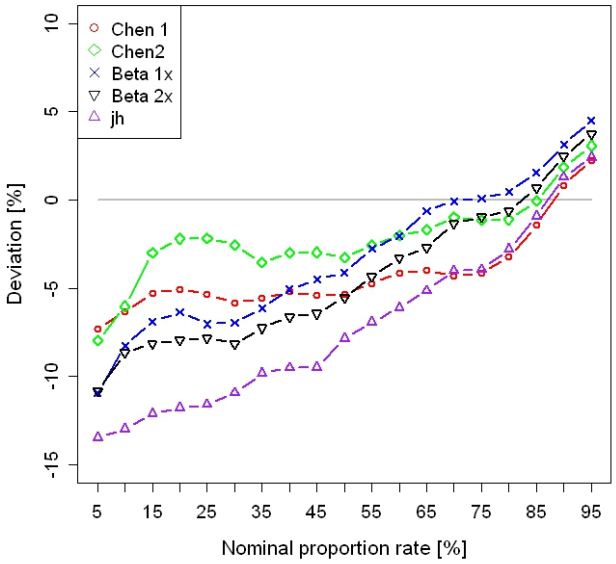

Fig. 3-54 Calibration diagram using QC estimator, for the offline test with WFA dataset A.

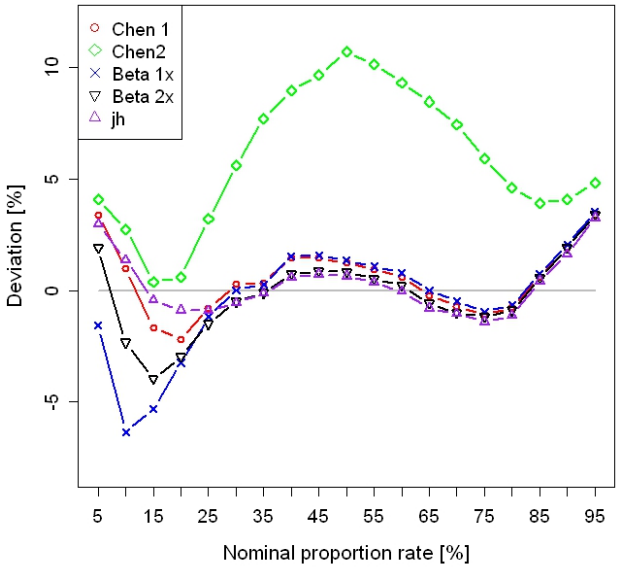

Fig. 3-55 Calibration diagram using NW estimator, for the offline test with WFB dataset A.

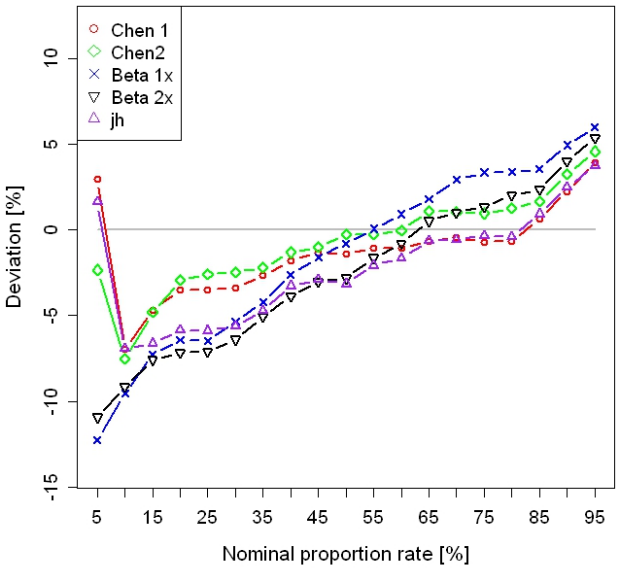

Fig. 3-56 Calibration diagram using QC estimator, for the offline test with WFB dataset A.

For this kernel size choice, only the *Boundary* kernel is not able to deliver results. Fig. 3-53 and Fig. 3-54 depict the calibration obtained for WFA using NW and QC estimators, respectively. In the former, *Chen 1* performs better than do the other kernels; however, in QC *Beta 1x* becomes a better performer for quantiles above 60%. In both the NW and QC estimators, the *jh* kernel has the worst performance, although in NW, *Chen 2* has also bad results. For this wind farm, there is a tendency to underestimate the quantiles.

As for WFB, Fig. 3-55 shows that for the NW estimator, *Chen 1* is the kernel with best overall calibration, and *Chen 2* is the worst. *Chen 2* becomes better when using QC between quantile 20%–60%, as depicted in Fig. 3-56.

Figs. 3-57 through 3-60 present the results for sharpness.

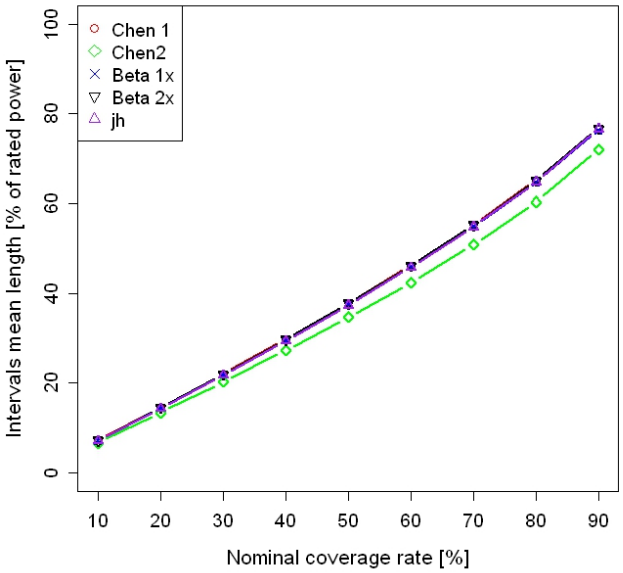

Fig. 3-57 Sharpness diagram using NW estimator, for the offline test with WFA dataset A.

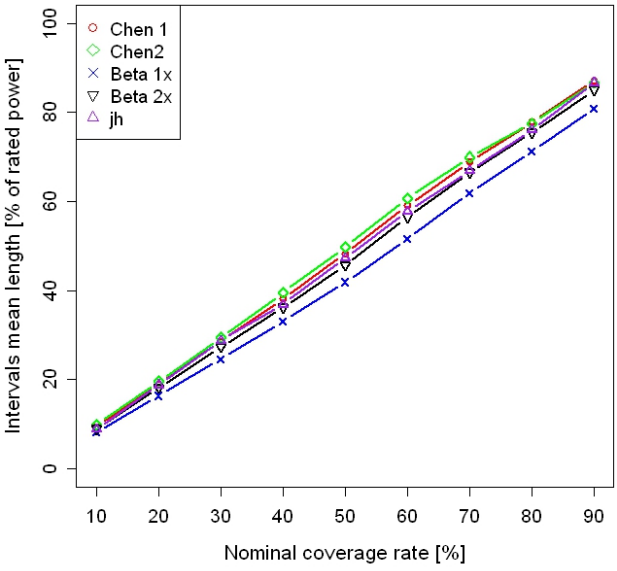

Fig. 3-58 Sharpness diagram using QC estimator, for the offline test with WFA dataset A.

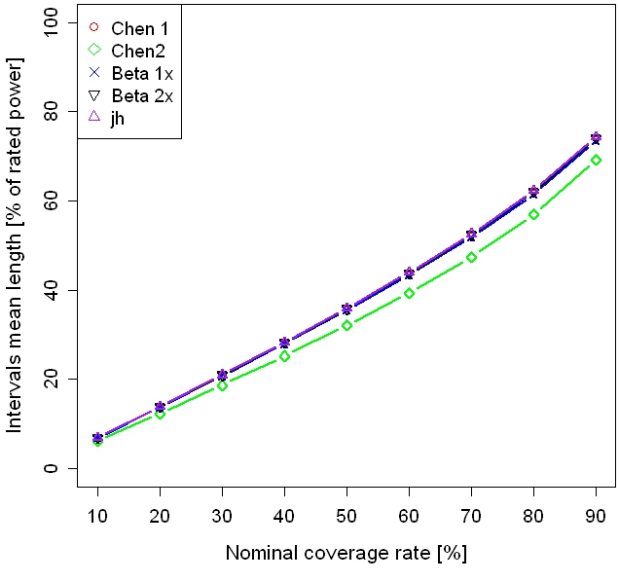

Fig. 3-59 Sharpness diagram using NW estimator, for the offline test with WFB dataset A.

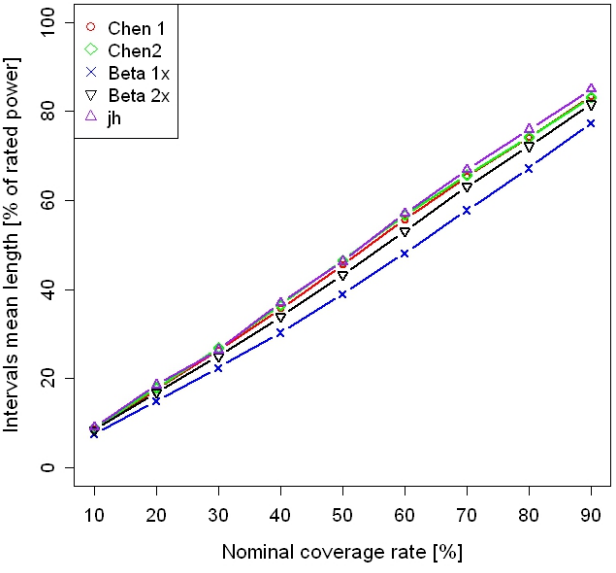

Fig. 3-60 Sharpness diagram using QC estimator, for the offline test with WFB dataset A.

In both wind farms, the sharpness and resolution results (Figs. 3-61 through 3-64) are the same. Using the NW estimator, *Chen 2* presents better results in terms of sharpness and resolution. As illustrated in Fig. 3-58 and Fig. 3-60, *Beta 1x* performs better in terms of sharpnes in both wind

farms when using the QC estimator. However, resolution results using QC are the same for all kernels in both wind farms, as depicted in Fig. 3-62 and Fig. 3-64.

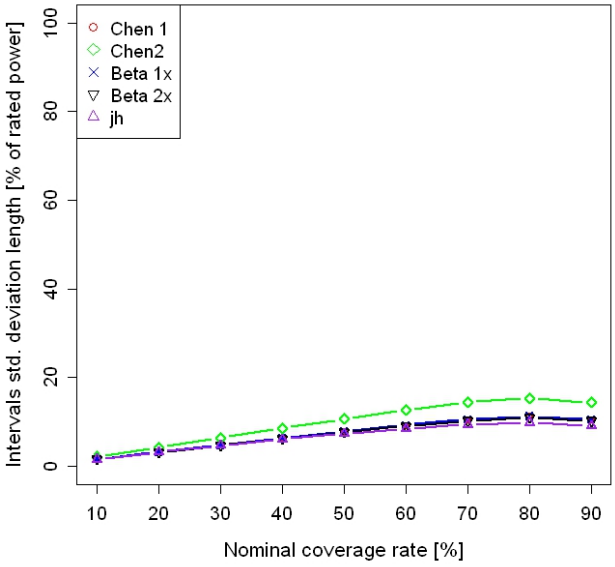

Fig. 3-61 Resolution diagram using the estimator, for the offline test with WFA dataset A.

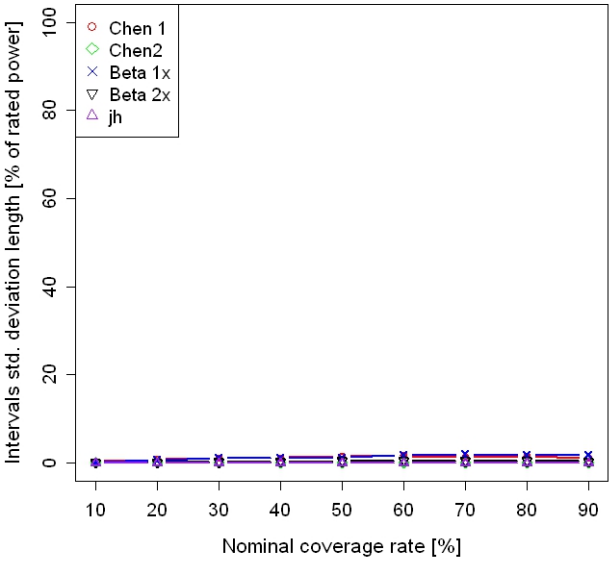

Fig. 3-62 Resolution diagram using QC estimator, for the offline test with WFA dataset A.

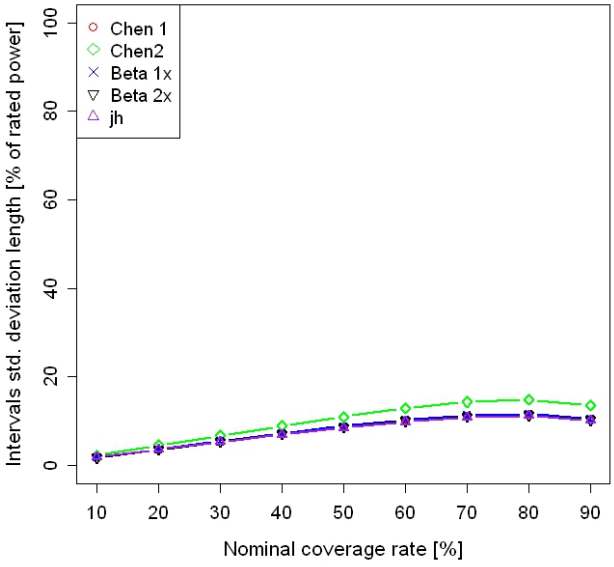

Fig. 3-63 Resolution diagram using NW estimator, for the offline test with WFB dataset A.

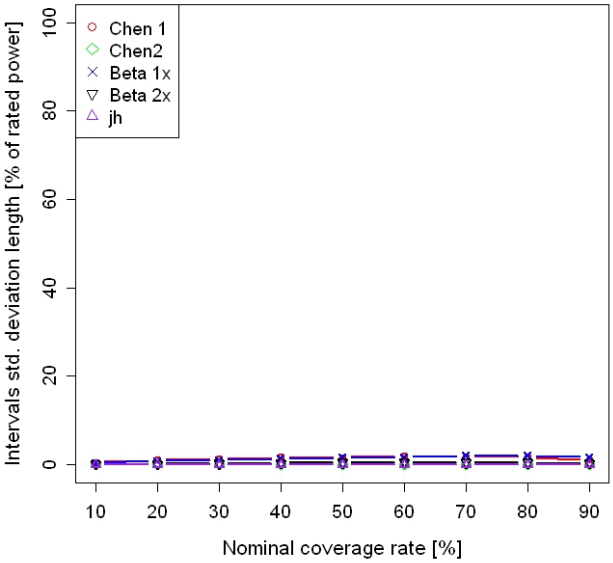

Fig. 3-64 Resolution diagram using QC estimator, for the offline test with WFB dataset A.

Figs. 3-65 through 3-68 present results for the skill score.

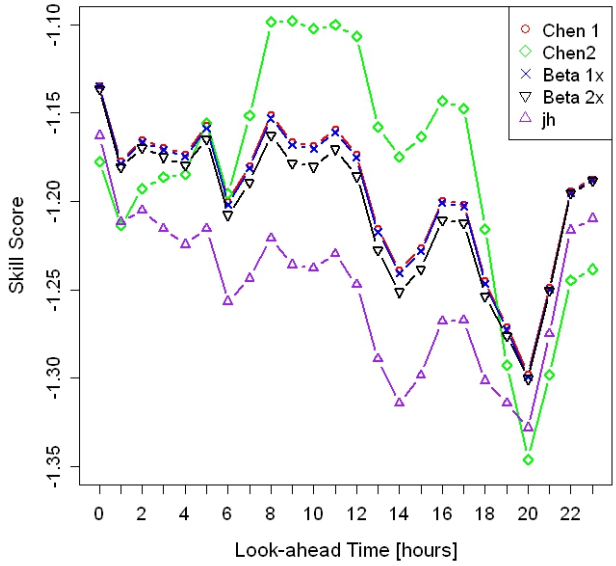

Fig. 3-65 Skill score diagram using NW estimator, for the offline test with WFA dataset A.

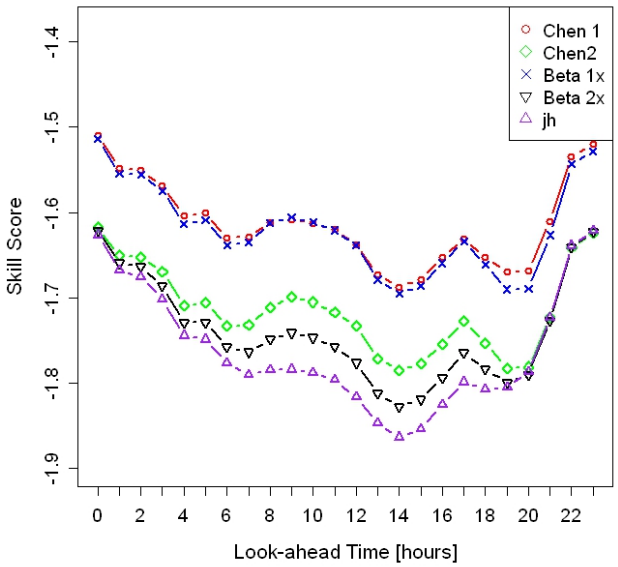

Fig. 3-66 Skill score diagram using QC estimator, for the offline test with WFA dataset A.

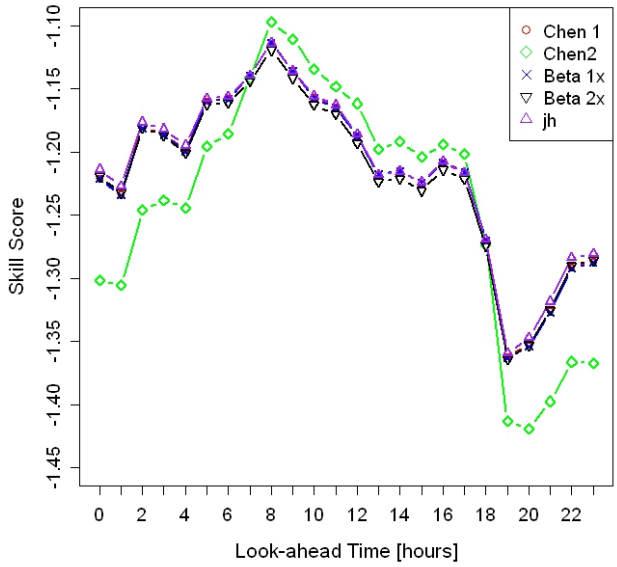

Fig. 3-67 Skill score diagram using NW estimator, for the offline test with WFB dataset A.

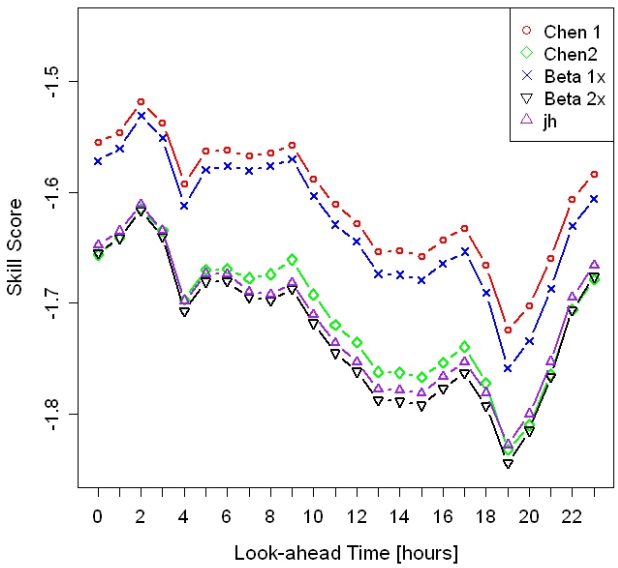

Fig. 3-68 Skill score diagram using QC estimator, for the offline test with WFB dataset A.

When using QC estimator, Fig. 3-66 and Fig. 3-68 show that, for both wind farms, *Chen 1* and *Beta 1x* have the best performance in terms of skill score, although the former is better; on the other hand, *jh* has the worst skill score. As depicted in Fig. 3-65 and Fig. 3-67, when using the NW estimator in both wind farms, the *Chen 2* kernel has the worst performance in terms of the skill score in the first and last look-ahead hours, this being the best kernel during the hours in between.

The main conclusions for this kernel choice are that *Chen 1* has the best overall calibration, and there is a tendency to underestimate the quantiles. In terms of sharpness and resolution, *Chen 2* has the best performance when using the NW estimator. Even though all of the kernels have the same resolution, when using QC, the best sharpness results are performed by *Beta 1x*. As for the skill score, *jh* has the worst overall performance for QC in both wind farms, and *Chen 1* presents the best performance. In addition to this result, for both wind farms when using NW, *Chen 2* has the worst performance in terms of the skill score in the first and last look-ahead hours, this being the best kernel during the hours in between.

Kernel Sizes: $(h_{Power}; h_{WindSpeed}) = (0.01; 1.2)$

Figs. 3-69 through 3-72 show results of offline testing of calibration at the stated kernel sizes using the WFA and WFB datasets.

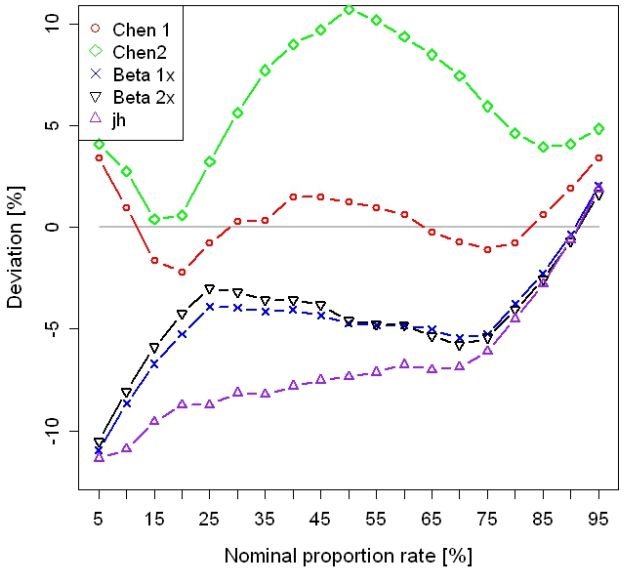

Fig. 3-69 Calibration diagram using NW estimator, for the offline test with WFA dataset A.

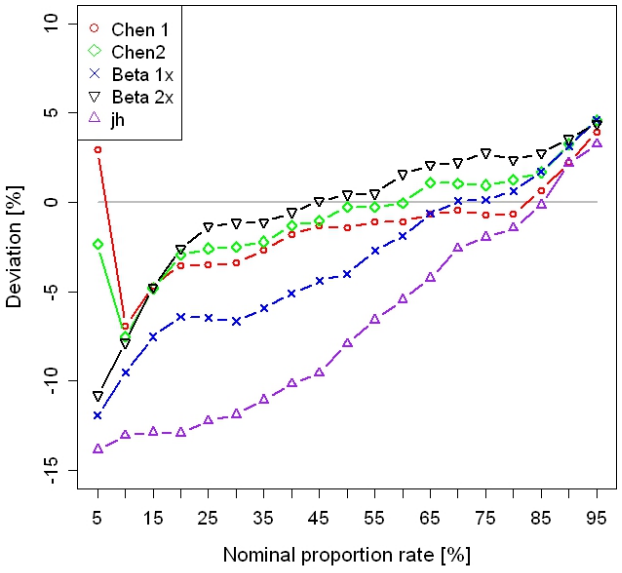

Fig. 3-70 Calibration diagram using QC estimator, for the offline test with WFA dataset A.

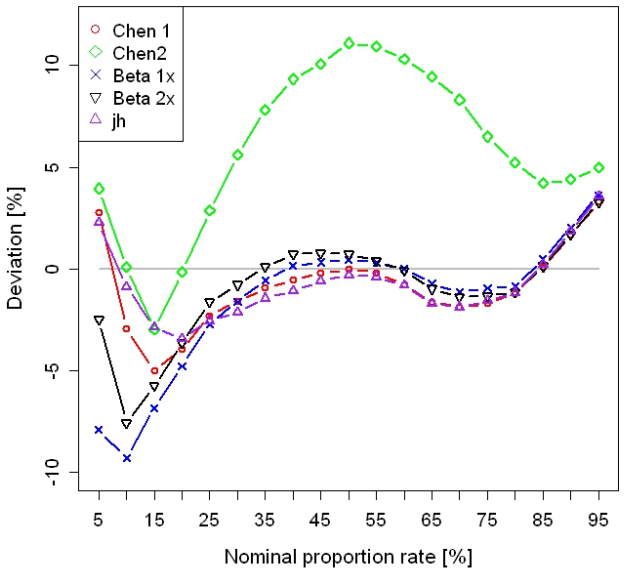

Fig. 3-71 Calibration diagram using NW estimator, for the offline test with WFB dataset A.

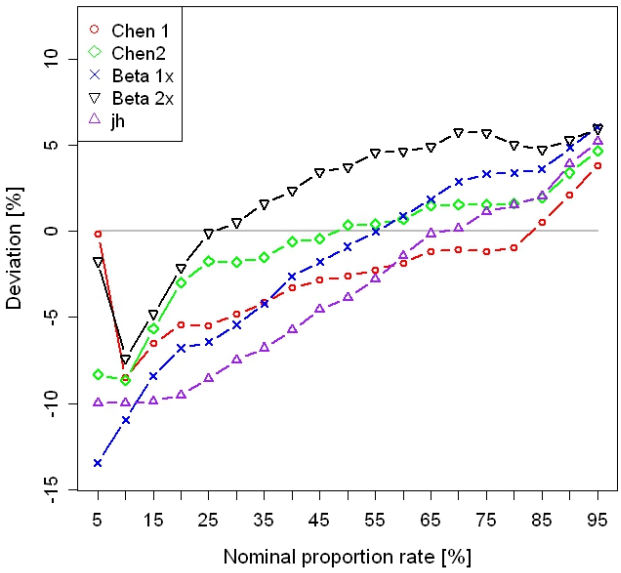

Fig. 3-72 Calibration diagram using QC estimator, for the offline test with WFB dataset A.

Using the NW estimator, *Chen 2* presents better results in terms of sharpness (Figs. 3-73 through 3-76) and resolution (Figs. 3-77 through 3-80). As illustrated in Fig. 3-74 and Fig. 3-76, *Beta 1x* performs better in terms of sharpness in both wind farms when using the QC estimator. However, resolution results using QC are similar for all kernels in both wind farms as depicted in Fig. 3-78 and Fig. 3-80.

Similar to the previous case, for this kernel size choice, only the *Boundary* kernel is not able to deliver results. Fig. 3-69 and Fig. 3-70 depict the calibration obtained for WFA using the NW and QC estimators, respectively. In the former, *Chen 1* performs better than do the other kernels; however, in QC, *Chen 2* is better for quantiles between 15% and 55%. In both the NW and QC estimators, the *jh* kernel has the worst overall performance, although in NW, *Chen 2* pulls bad results as well.

As for WFB, Fig. 3-71 shows that for the NW estimator, *Chen 1* is the kernel with best overall calibration, and *Chen 2* is the worst. *Chen 2* becomes a better performer when using QC, as depicted in Fig. 3-72. For both wind farms, there is a tendency to underestimate the quantiles.

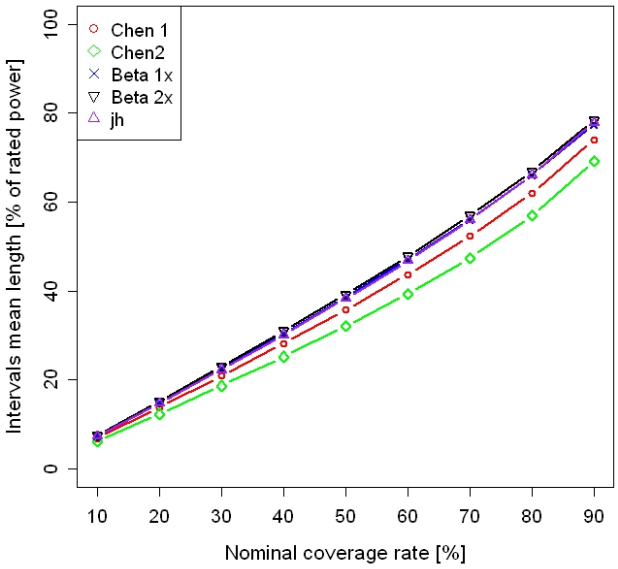

Fig. 3-73 Sharpness diagram using NW estimator, for the offline test with WFA dataset A.

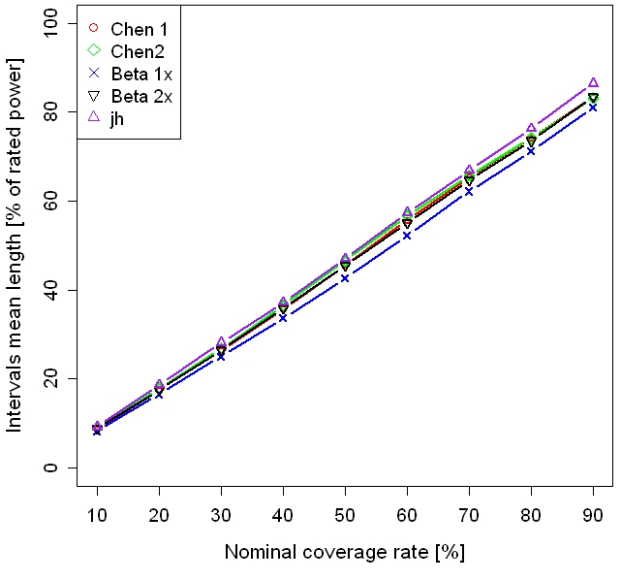

Fig. 3-74 Sharpness diagram using QC estimator, for the offline test with WFA dataset A.

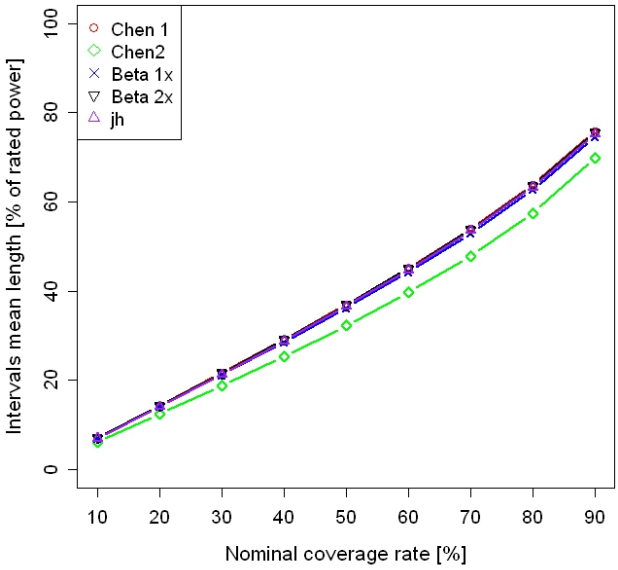

Fig. 3-75 Sharpness diagram using NW estimator, for the offline test with WFB dataset A.

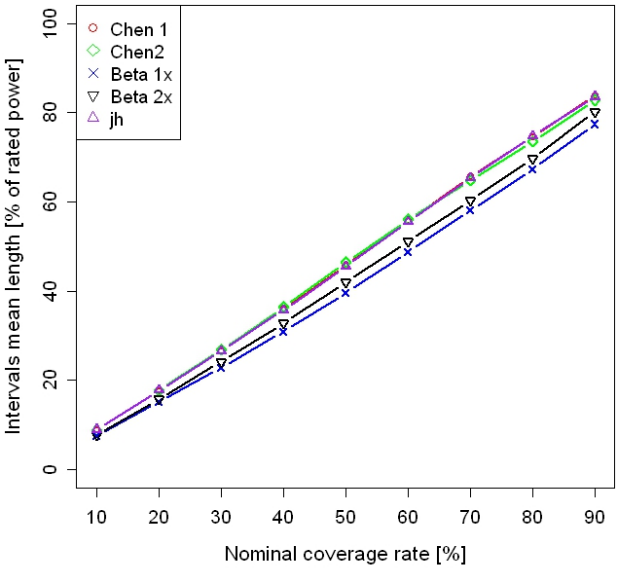

Fig. 3-76 Sharpness diagram using QC estimator, for the offline test with WFB dataset A.

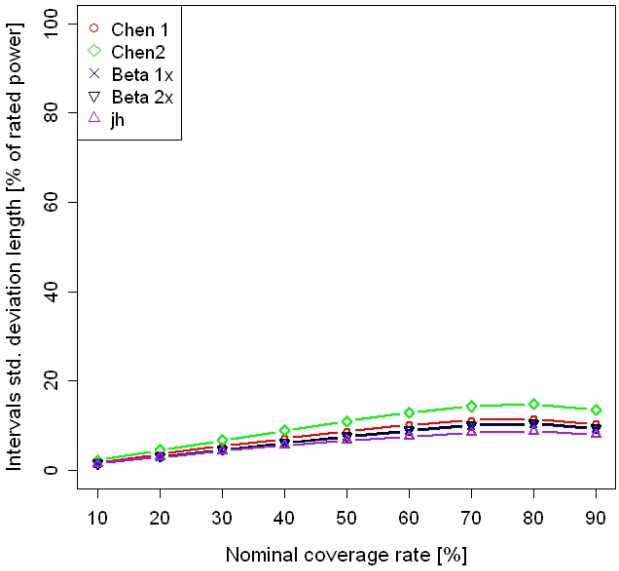

Fig. 3-77 Resolution diagram using the estimator, for the offline test with WFA dataset A.

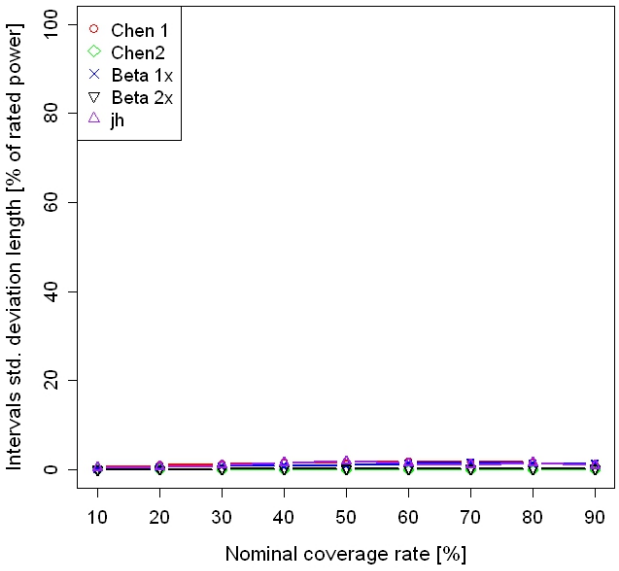

Fig. 3-78 Resolution diagram using QC estimator, for the offline test with WFA dataset A.

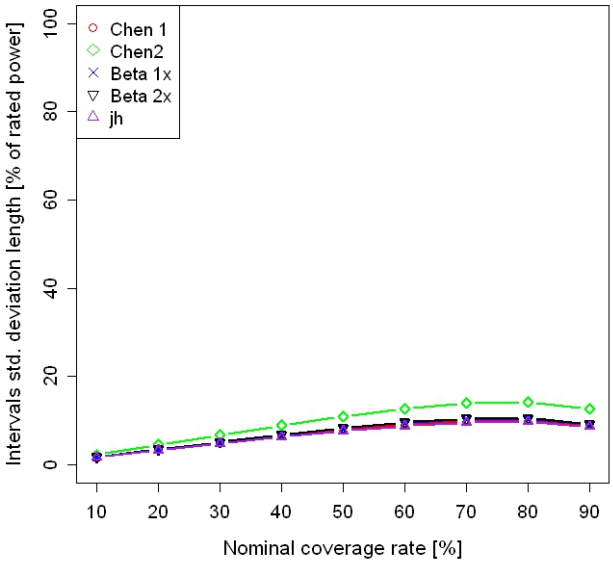

Fig. 3-79 Resolution diagram using NW estimator, for the offline test with WFB dataset A.

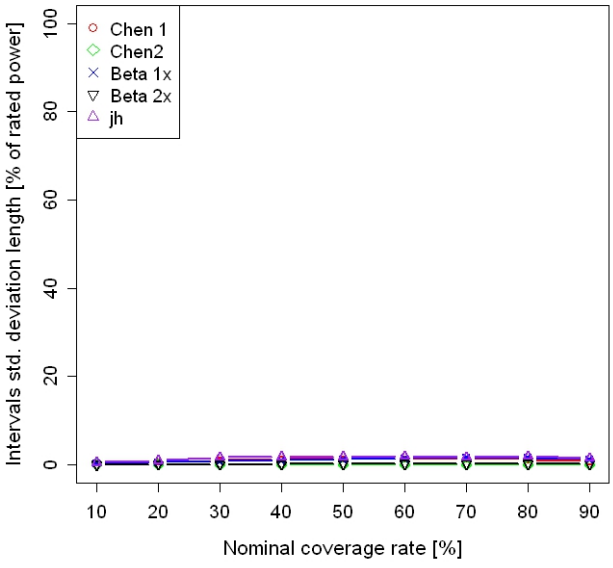

Fig. 3-80 Resolution diagram using QC estimator, for the offline test with WFB dataset A.

In terms of the skill score (Figs. 3-81 through 3-84), when using the QC estimator, Fig. 3-82 and Fig. 3-84 show that for both wind farms, *Chen 1* has the best performance; on the other hand, *jh*

has the worst skill score. As depicted in Fig. 3-81 and Fig. 3-83, when using the NW estimator in both wind farms, the *Chen 2* kernel has the worst performance in terms of the skill score in the first and last look-ahead hours, this being the best kernel during the hours in between.

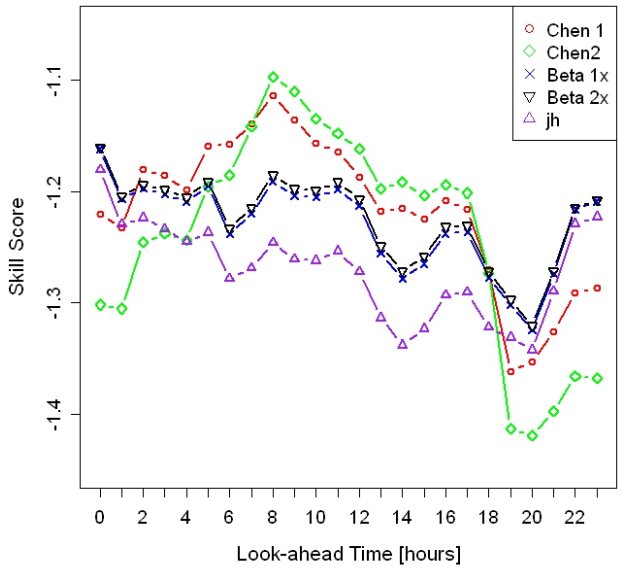

Fig. 3-81 Skill score diagram using NW estimator, for the offline test with WFA dataset A.

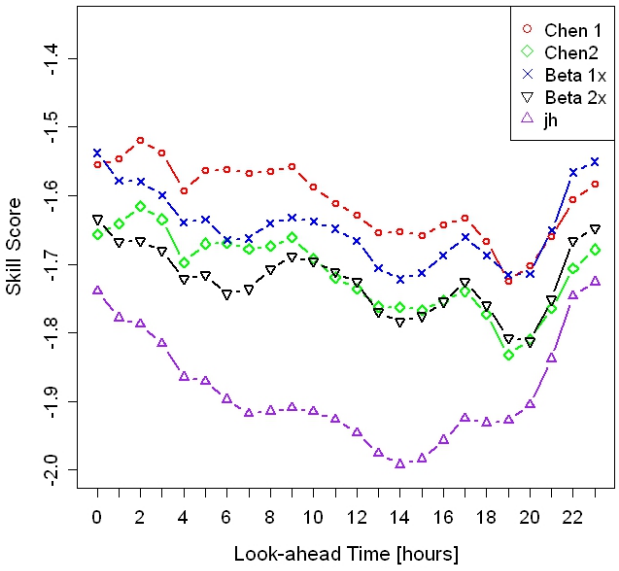

Fig. 3-82 Skill score diagram using QC estimator, for the offline test with WFA dataset A.

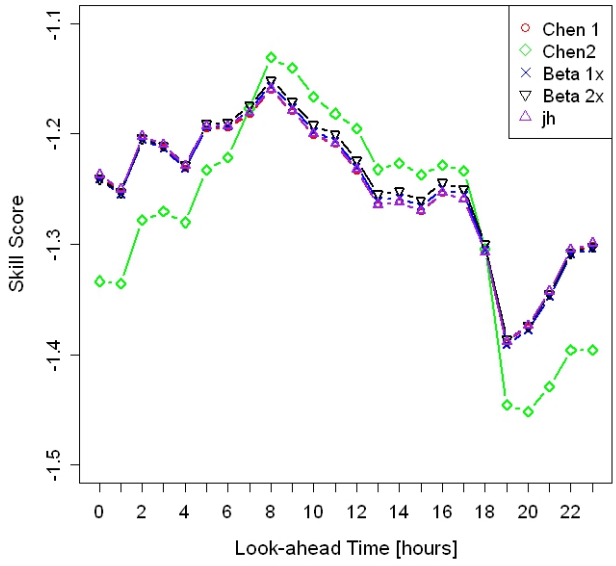

Fig. 3-83 Skill score diagram using NW estimator, for the offline test with WFB dataset A.

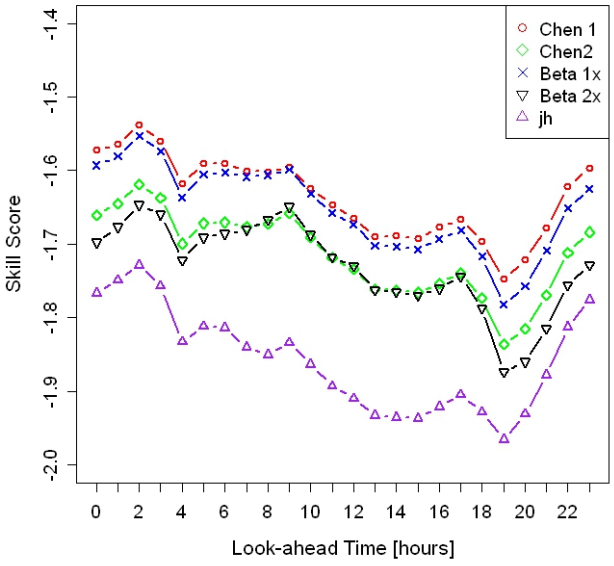

Fig. 3-84 Skill score diagram using QC estimator, for the offline test with WFB dataset A.

The main conclusions for this kernel are similar to the ones drawn in the previous case. Hence, *Chen 1* has the best overall calibration, and there is a tendency to underestimate the quantiles. In terms of sharpness and resolution, *Chen 2* has the best performance when using the NW estimator. Even though all of the kernels have the same resolution, when using QC, the best sharpness results are performed by *Beta 1x*. As for the skill score, *jh* has the worst overall performance for QC in both wind farms, and *Chen 1* presents the best performance. In addition to this finding, for both wind farms when using NW, *Chen 2* has the worst performance in terms of the skill score in the first and last look-ahead hours, this being the best kernel during the hours in between.

Kernel Method Results

Results comparing different kernels under distinct kernel sizes proved that Chen 1 (gamma and kernel function) is the estimator with the best overall calibration performance. All kernels have similar sharpness results — except when the power kernel size is large, when Chen 2 and Beta 1x are good methods using NW and QC, respectively. Similar results among kernels are also obtained for resolution, although Chen 2 and Boundary stand out for having good performance. In terms of skill score, Chen 1 is better for the QC estimator, and for NW, Chen 2 performs better during mid look-ahead hours.

3.4.3.3 Evaluation of Dataset Characteristics Sensitivity

Comparisons were performed for six different datasets:

A. Training period from January 2, 2009, until July 31, 2009; and testing period between August 1, 2009, and February 20, 2010.
B. Training period from August 1, 2009, until February 20, 2010; and testing period between January 2, 2009, and July 31, 2009.
C. Training period between January 2, 2009, and June 30, 2009; and testing period from July 1, 2009, until February 20, 2010.
D. Training period between July 1, 2009, and February 20, 2010; and testing period from January 2, 2009, until June 20, 2009.
E. Training period from January 2, 2009, until August 31, 2009; and testing period between September 1, 2009, and February 20, 2010.
F. Training period from September 1, 2009, until February 20, 2010; and testing period between January 2, 2009, and August 31, 2009.

Because from all of the kernels used, *Chen 1* was the one which revealed the best overall performance (namely in terms of sharpness), it is the one chosen to evaluate the sensitivity of the results toward the dataset characteristics. Hence, for a kernel size choice of (σ_{Power}; $\sigma_{WindSpeed}$) = (0.002; 0.04), results are presented for datasets A–F in both wind farm A (WFA) and B (WFB).

Dataset A

The main characteristics of the training and testing dataset A of WFA are presented in Table 3-2 and for WFB are presented in Table 3-3. The training period was from January 2, 2009, until July 31, 2009; the testing period was between August 1, 2009, and February 20, 2010.

The statistical field that changes the most in this dataset is kurtosis. Wind speed kurtosis doubles from training to testing for both WFA and WFB, and the inverse occurs for power in WFA. The values of power kurtosis are negative, whereas they are positive for wind speed. Moreover, in WFB, the training period is slightly longer than the testing period.

Table 3-2 Statistical characteristics of the WFA training and testing dataset A.

Variables	Nº Points	Mean	Median	Std. Dev.	Skewness	Kurtosis	IQR[a]
Train Dataset							
Wind Speed (m/s)	4992	7.488	7.045	3.370	0.552	0.055	4.526
Wind power (p.u.)	4992	0.328	0.259	0.273	0.485	-1.012	0.468
Test Dataset							
Wind Speed (m/s)	4680	6.981	6.626	3.288	0.555	0.119	4.434
Wind power (p.u.)	4680	0.315	0.235	0.293	0.727	-0.618	0.471

[a] IQR = Inter-quantile range; p.u. = power unit.

Table 3-3 Statistical characteristics of the WFB training and testing dataset A.

Variables	Nº Points	Mean	Median	Std. Dev.	Skewness	Kurtosis	IQR[a]
Train Dataset							
Wind Speed (m/s)	5052	7.299	6.892	3.418	0.570	0.060	4.526
Wind power (p.u.)	5052	0.302	0.238	0.260	0.606	-0.843	0.432
Test Dataset							
Wind Speed (m/s)	4824	6.661	6.253	3.308	0.624	0.192	4.548
Wind power (p.u.)	4824	0.309	0.227	0.290	0.768	-0.559	0.463

[a] IQR = Inter-quantile range; p.u. = power unit.

Fig. 3-85 depicts the calibration obtained for WFA using various estimators. This graph shows that QR approaches (linear and splines) have the best overall calibration results, with KDF estimators (NW and QC) performing better than the splines QR only for quantiles above 65%. KDF estimators underestimate the quantiles.

Fig. 3-86 presents sharpness results, which are better for splines QR and worse for QC. These estimators have the best resolution as well, as depicted in Fig. 3-87.

The splines QR has better performance in terms of the skill score, whereas linear QR has the worst. Hence, the skill score performance of the KDF estimators lies between the QR approaches, where NW is better than QC, as shown in Fig. 3-88.

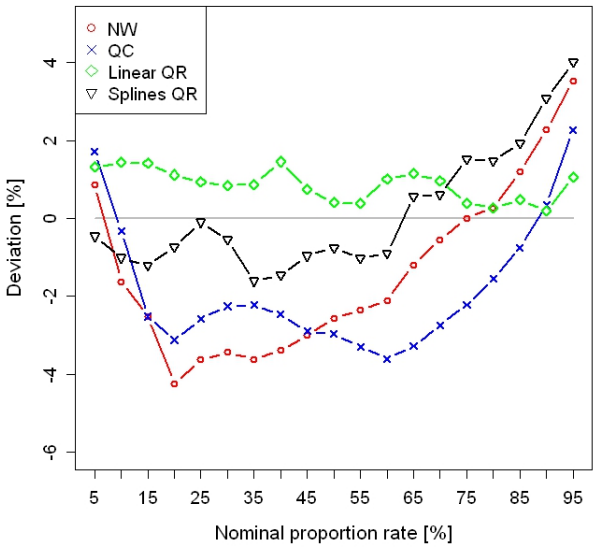

Fig. 3-85 Calibration diagram for the offline test with WFA dataset A.

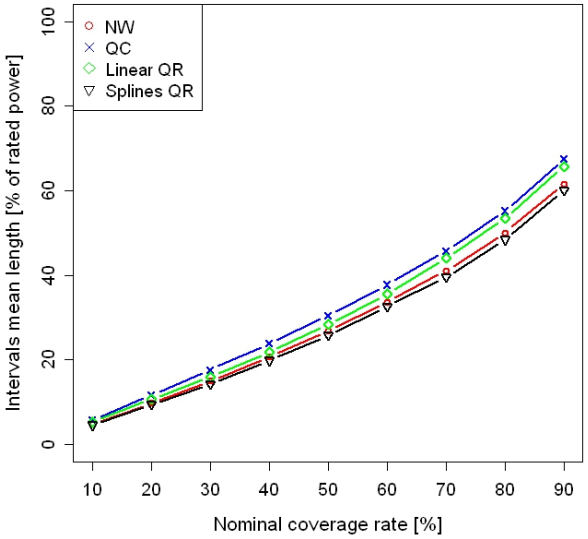

Fig. 3-86 Sharpness diagram for the offline test with WFA dataset A.

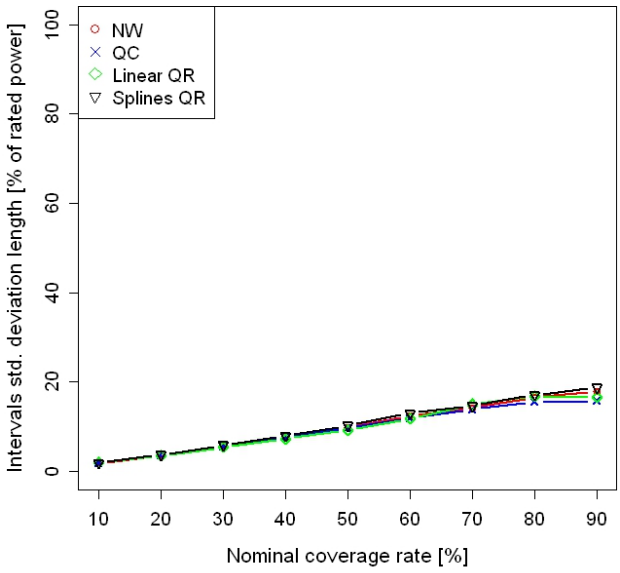

Fig. 3-87 Resolution diagram for the offline test with WFA dataset A.

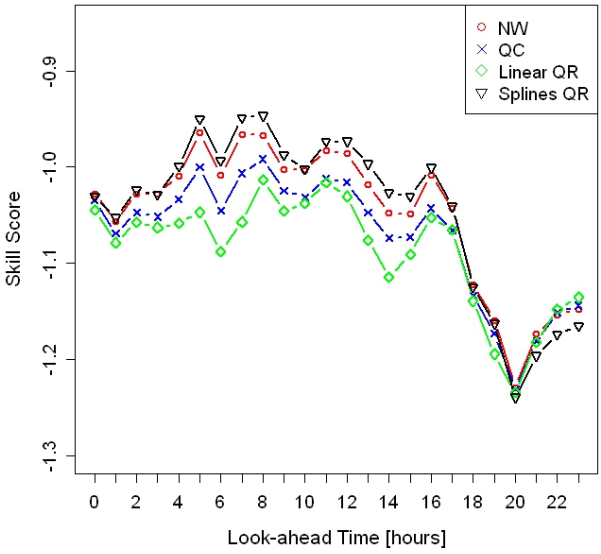

Fig. 3-88 Skill score diagram for the offline test with WFA dataset A.

Fig. 3-89 through Fig. 3-92 depict the results for WFB. The graphs show that the behavior is similar to that of the WFA case, except for the calibration and resolution. In fact, for quantiles above 25%, the KDF estimators have better calibration performance than splines QR, with linear QR being the worst approach; moreover, quantiles tend to be overestimated. Linear QR has the worst resolution in WFB.

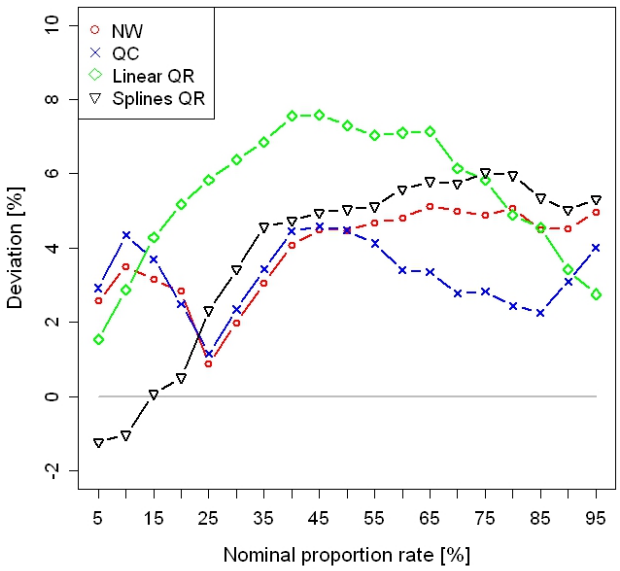

Fig. 3-89 Calibration diagram for the offline test with WFB dataset A.

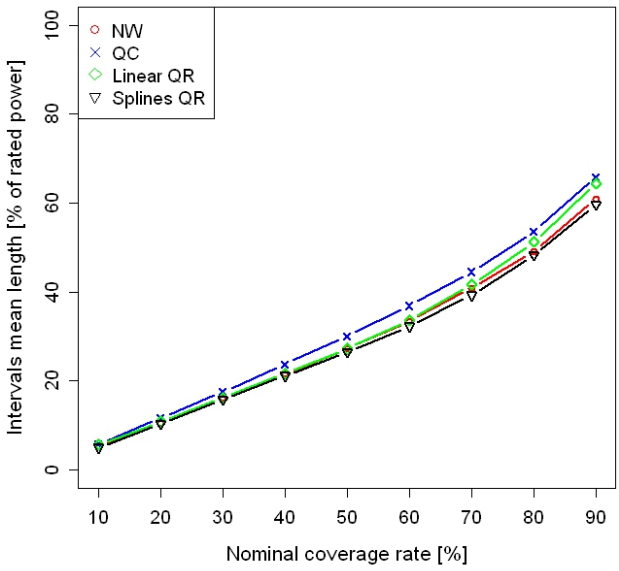

Fig. 3-90 Sharpness diagram for the offline test with WFB dataset A.

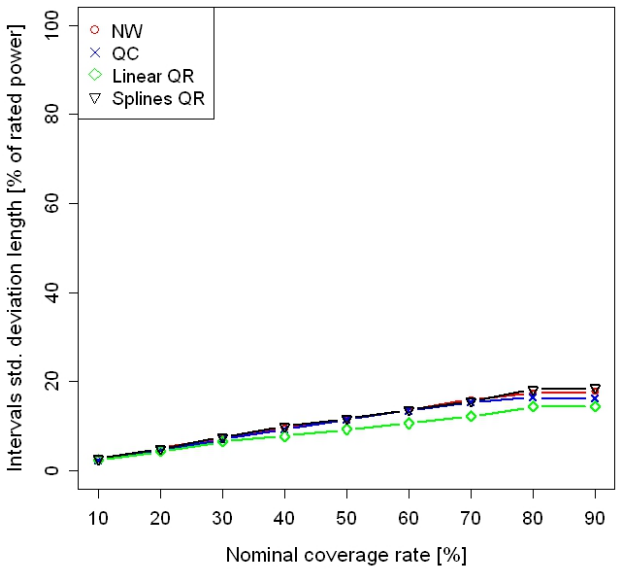

Fig. 3-91 Resolution diagram for the offline test with WFB dataset A.

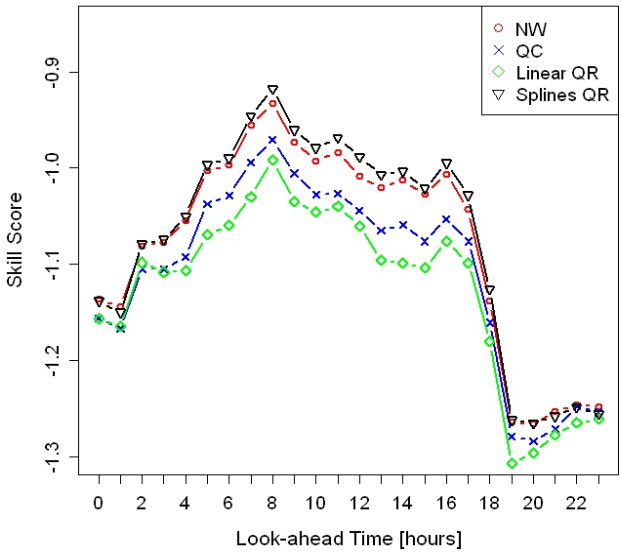

Fig. 3-92 Skill score diagram for the offline test with WFB dataset A.

The main conclusions for dataset A are that splines QR has the best overall calibration, although KDF approaches also present good performance, particularly in WFB. Splines QR has the best sharpness in both wind farms, and linear QR the worst resolution in WFB. In terms of skill score,

QR approaches have the best and the worst performance and, among the KDF estimators, NW is better.

Dataset B

The main characteristics of the training and testing dataset B of WFA are presented in Table 3-4 and for WFB are presented in Table 3-5. The training period was between August 1, 2009, and February 20, 2010; the testing period was from January 2, 2009, until July 31, 2009.

The statistical field that changes most in this dataset is kurtosis. Because the only difference between this dataset and the previous is the exchange of training and testing, the changes in kurtosis referred to for dataset A are inverted in dataset B. Hence, wind speed kurtosis halves from training to testing for both WFA and WFB. In contrast to wind speed, power kurtoses are negative, and in WFA their values double from training to testing. Moreover, in WFB, the training period is slightly shorter than the testing period.

Table 3-4 Statistical characteristics of the WFA training and testing dataset B.

Variables	Nº Points	Mean	Median	Std. Dev.	Skewness	Kurtosis	IQR[a]
Train Dataset							
Wind Speed (m/s)	4680	6.981	6.626	3.288	0.555	0.119	4.434
Wind power (p.u.)	4680	0.315	0.235	0.293	0.727	-0.618	0.471
Test Dataset							
Wind Speed (m/s)	4992	7.488	7.045	3.370	0.552	0.055	4.526
Wind power (p.u.)	4992	0.328	0.259	0.273	0.485	-1.012	0.468

[a] IQR = Inter-quantile range; p.u. = power unit.

Table 3-5 Statistical characteristics of the WFB training and testing dataset B.

Variables	Nº Points	Mean	Median	Std. Dev.	Skewness	Kurtosis	IQR[a]
Train Dataset							
Wind Speed (m/s)	4824	6.661	6.253	3.308	0.624	0.192	4.548
Wind power (p.u.)	4824	0.309	0.227	0.290	0.768	-0.559	0.463
Test Dataset							
Wind Speed (m/s)	5052	7.299	6.892	3.418	0.570	0.060	4.526
Wind power (p.u.)	5052	0.302	0.238	0.260	0.606	-0.843	0.432

[a] IQR = Inter-quantile range; p.u. = power unit.

Fig. 3-93 depicts the calibration obtained for WFA using various estimators. This graph shows that for quantiles below 60%, QC has the best calibration results, and above 60%, the QR estimators become better. Quantiles tend to be underestimated.

Fig. 3-94 presents sharpness results, which are better for splines QR and worse for QC. In terms of resolution, linear QR performs better than do the other estimators, which are almost the same, as depicted in Fig. 3-95.

Fig. 3-96 shows that QC has better performance in terms of the skill score, while linear QR has the worst.

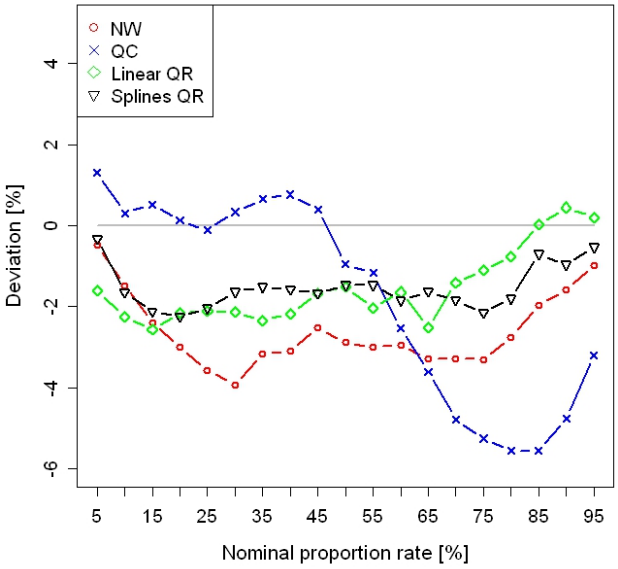

Fig. 3-93 Calibration diagram for the offline test with WFA dataset B.

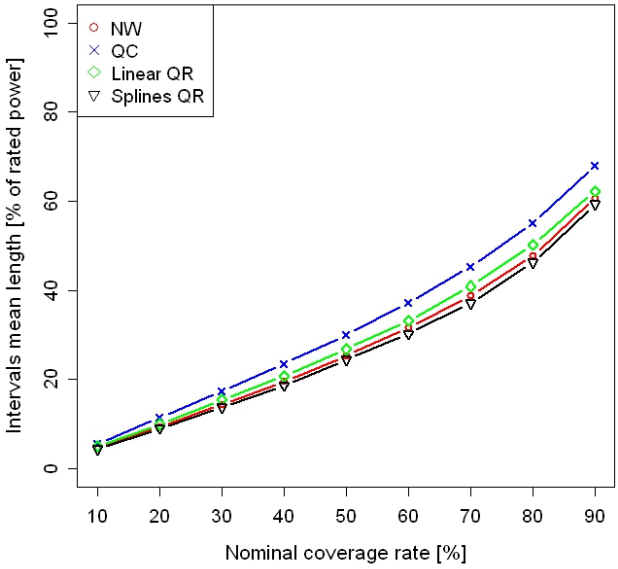

Fig. 3-94 Sharpness diagram for the offline test with WFA dataset B.

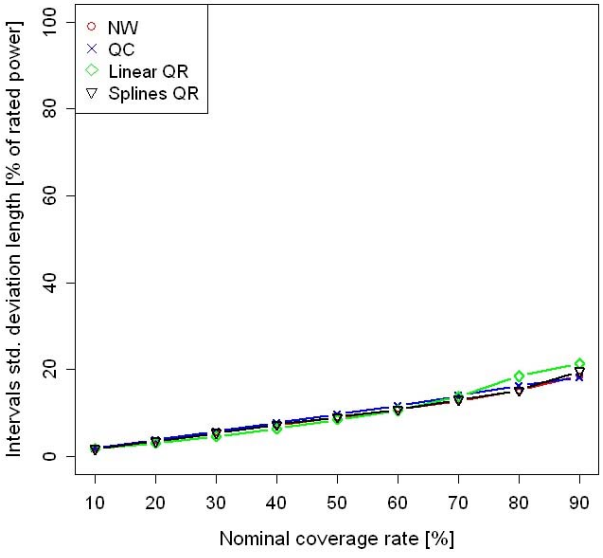

Fig. 3-95 Resolution diagram for the offline test with WFA dataset B.

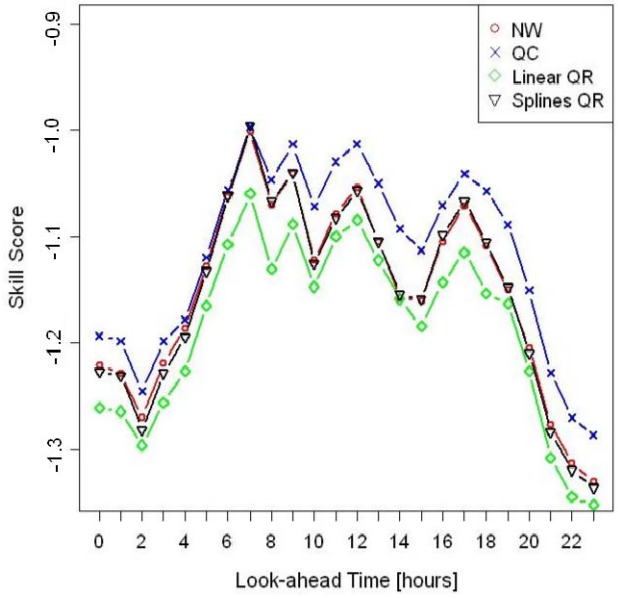

Fig. 3-96 Skill score diagram for the offline test with WFA dataset B.

Fig. 3-97 through Fig. 3-100 depict the results for WFB. The graphs show that the behavior is similar to that found for WFA.

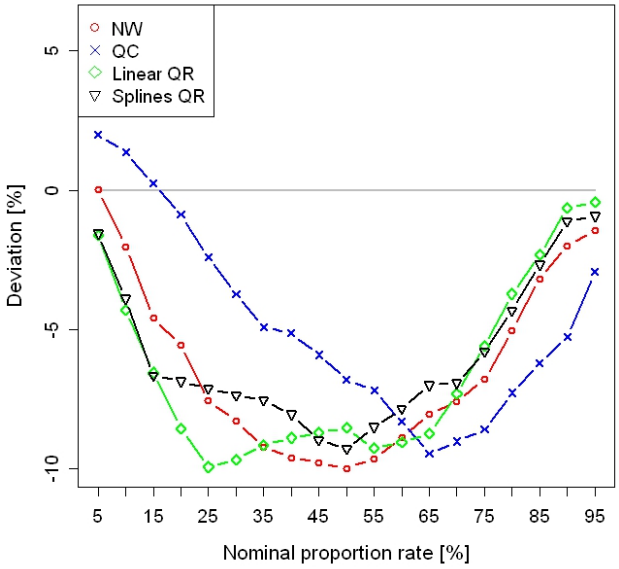

Fig. 3-97 Calibration diagram for the offline test with WFB dataset B.

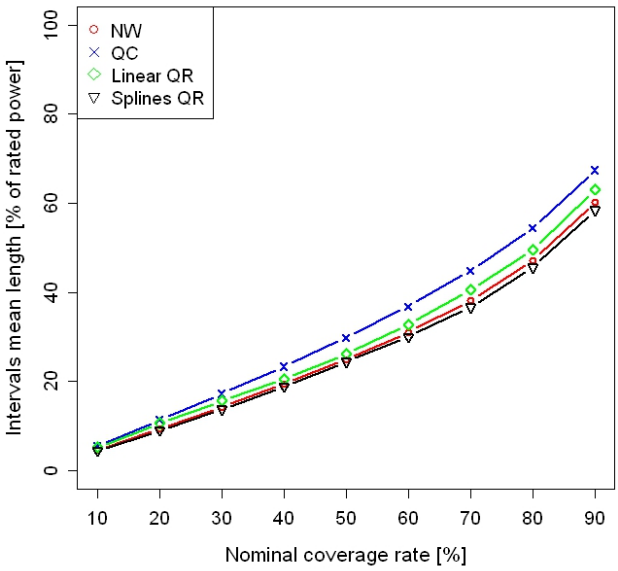

Fig. 3-98 Sharpness diagram for the offline test with WFB dataset B.

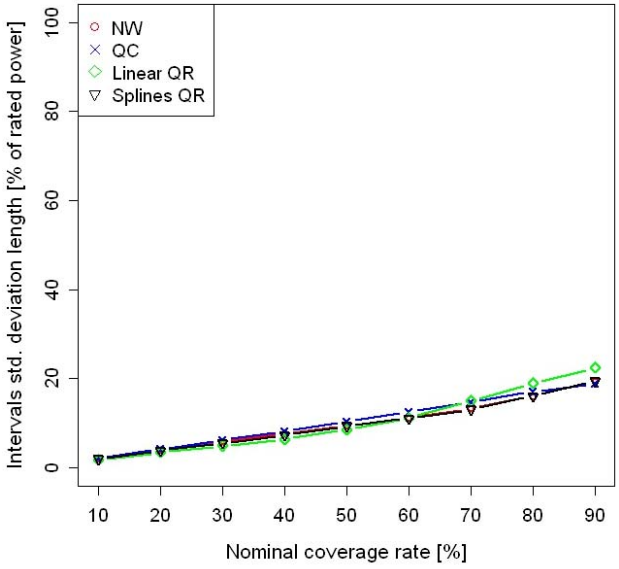

Fig. 3-99 Resolution diagram for the offline test with WFB dataset B.

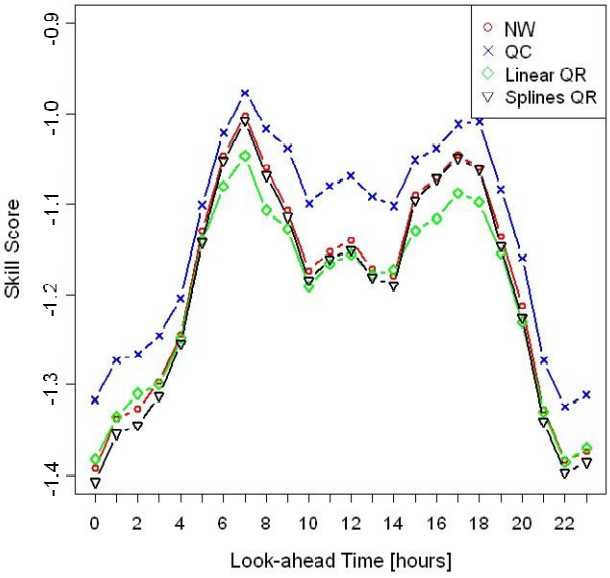

Fig. 3-100 Skill score diagram for the offline test with WFB dataset B.

The main conclusions for dataset B are that QC has the best calibration for quantiles below 60%, and estimators underestimate the quantiles; splines QR has the best sharpness and linear QR the best resolution; in terms of skill score QC has the best performance.

Dataset C

The main characteristics of the training and testing datasets C of WFA are presented in Table 3-6, and for WFB are presented in Table 3-7. The training period was between January 2, 2009, and June 30, 2009; and the testing period was from July 1, 2009, until February 20, 2010.

The statistical field that changes most in this dataset is kurtosis. Wind speed kurtosis changes sign from negative to positive between training and testing, respectively. In WFA, the module of wind speed kurtosis increases 6 times, while in WFB, it increases 22 times. Power kurtoses are negative and halve their value from training to testing in both wind farms. Moreover, the training period is slightly shorter than the testing period.

Table 3-6 Statistical characteristics of the WFA training and testing dataset C.

Variables	Nº Points	Mean	Median	Std. Dev.	Skewness	Kurtosis	IQR[a]
Train Dataset							
Wind Speed (m/s)	4248	7.739	7.297	3.410	0.515	-0.030	4.681
Wind power (p.u.)	4248	0.351	0.308	0.275	0.381	-1.115	0.482
Test Dataset							
Wind Speed (m/s)	5424	6.853	6.474	3.231	0.575	0.181	4.387
Wind power (p.u.)	5424	0.299	0.215	0.286	0.800	-0.468	0.463

[a] IQR = Inter-quantile range; p.u. = power unit.

Table 3-7 Statistical characteristics of the WFB training and testing dataset C.

Variables	Nº Points	Mean	Median	Std. Dev.	Skewness	Kurtosis	IQR[a]
Train Dataset							
Wind Speed (m/s)	4308	7.557	7.122	3.456	0.541	-0.009	4.770
Wind power (p.u.)	4308	0.321	0.262	0.263	0.524	-0.957	0.443
Test Dataset							
Wind Speed (m/s)	5568	6.547	6.102	3.253	0.632	0.229	4.511
Wind power (p.u.)	5568	0.294	0.209	0.284	0.836	-0.409	0.449

[a] IQR = Inter-quantile range; p.u. = power unit.

Fig. 3-101 depicts the calibration obtained for WFA using various estimators. This graph shows that QR approaches have the best overall calibration results, with linear QR performing better for quantiles below 65%.

Fig. 3-102 presents sharpness results that are better for splines QR and worse for QC. In terms of resolution, linear QR is the worst, as depicted in Fig. 3-103.

The splines QR has better performance in terms of the skill score, while linear QR has the worst. Hence, the skill score performance of KDF estimators lies between QR approaches, with NW performing better than QC, as shown in Fig. 3-104.

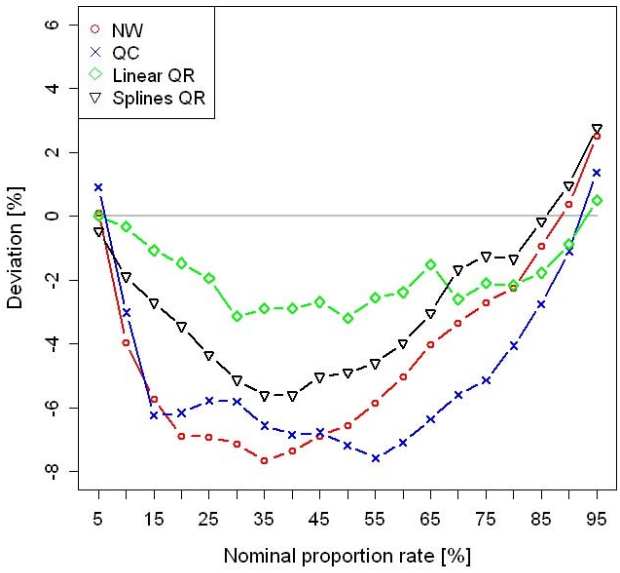

Fig. 3-101 Calibration diagram for the offline test with WFA dataset C.

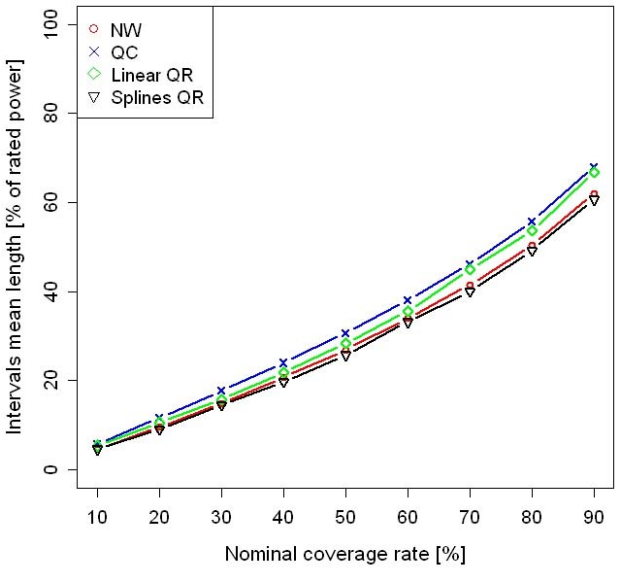

Fig. 3-102 Sharpness diagram for the offline test with WFA dataset C.

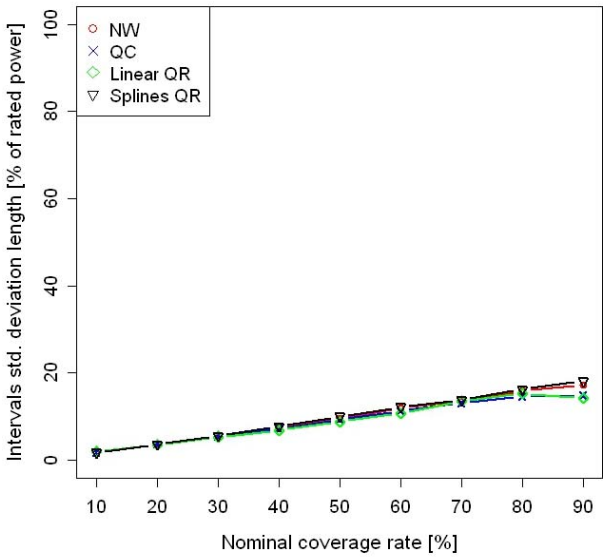

Fig. 3-103 Resolution diagram for the offline test with WFA dataset C.

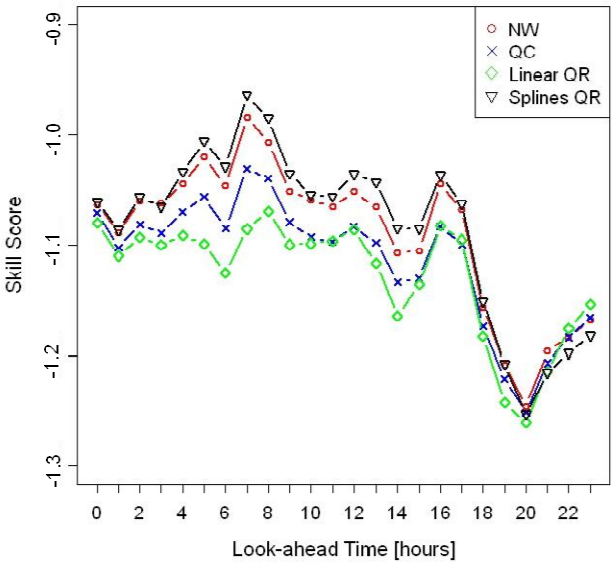

Fig. 3-104 Skill score diagram for the offline test with WFA dataset C.

Fig. 3-105 through Fig. 3-108 depict the results for WFB. The graphs show that the behavior is similar as in the WFA case, except for the calibration. In fact, for quantiles above 45%, KDF estimators become better performers than splines QR, with linear QR being the approach with the worst calibration performance. Moreover, estimators tend to overestimate the quantiles.

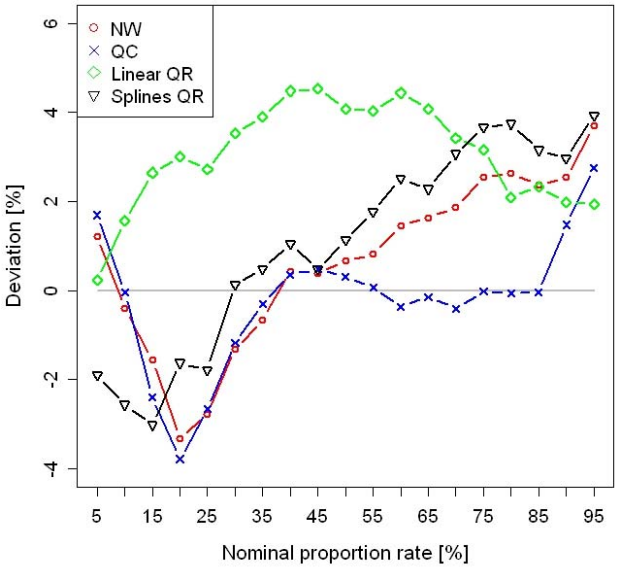

Fig. 3-105 Calibration diagram for the offline test with WFB dataset C.

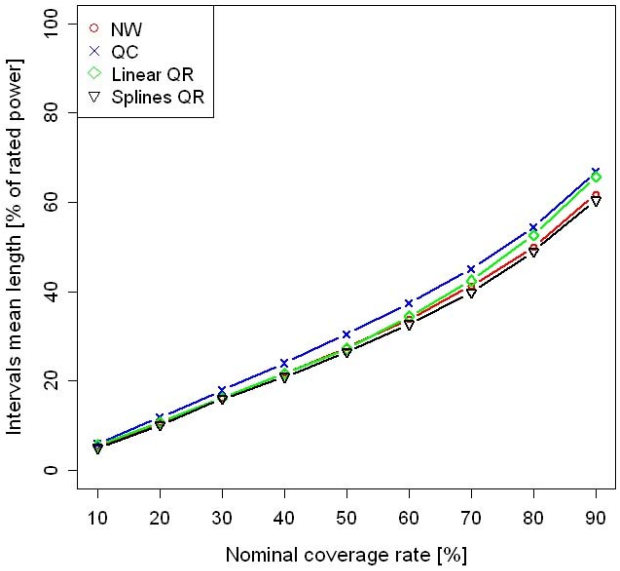

Fig. 3-106 Sharpness diagram for the offline test with WFB dataset C.

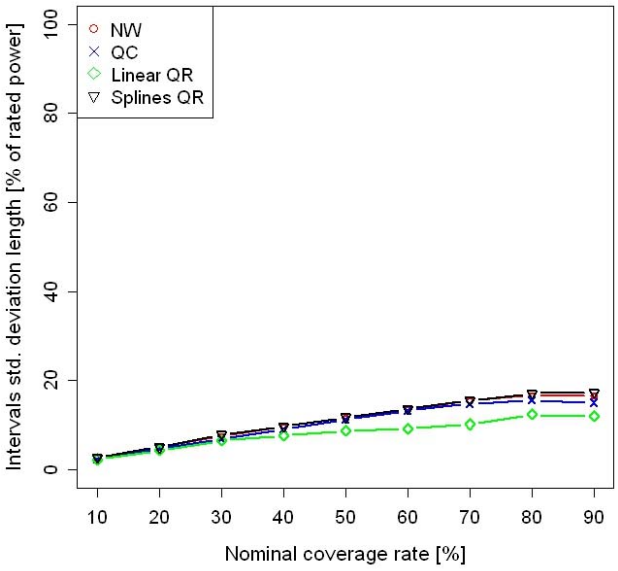

Fig. 3-107 Resolution diagram for the offline test with WFB dataset C.

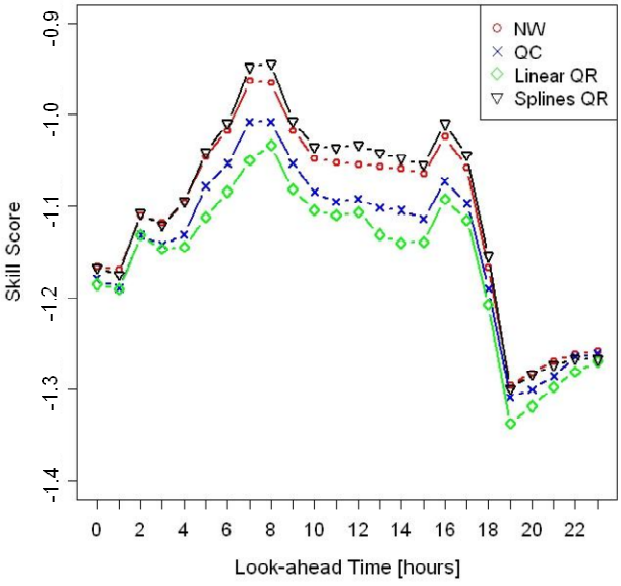

Fig. 3-108 Skill score diagram for the offline test with WFB dataset C.

The main conclusions for dataset C are that QR estimators have better overall calibration, although KDF approaches also present a good performance, particularly in WFB; splines QR has the best sharpness and linear QR the worst resolution; in terms of the skill score, QR approaches have the best and the worst performance and, among the KDF estimators, NW is better.

Dataset D

The main characteristics of the training and testing datasets D of WFA are presented in Table 3-8 and for WFB are presented in Table 3-9. The training period was from July 1, 2009, until February 20, 2010; and the testing period was between January 2, 2009, and June 30, 2009.

The statistical field that changes most in this dataset is kurtosis. Wind speed kurtosis changes sign from positive to negative between training and testing, respectively. In WFA, the module of wind speed kurtosis decreases 6 times, while in WFB, it decreases 22 times. Power kurtoses are negative and double their value from training to testing in both wind farms. Moreover, the training period is slightly longer than the testing one.

Table 3-8 Statistical characteristics of the WFA training and testing dataset D.

Variables	Nº Points	Mean	Median	Std. Dev.	Skewness	Kurtosis	IQR[a]
Train Dataset							
Wind Speed (m/s)	5424	6.853	6.474	3.231	0.575	0.181	4.387
Wind power (p.u.)	5424	0.299	0.215	0.286	0.800	-0.468	0.463
Test Dataset							
Wind Speed (m/s)	4248	7.739	7.297	3.410	0.515	-0.030	4.681
Wind power (p.u.)	4248	0.351	0.308	0.275	0.381	-1.115	0.482

[a] IQR = Inter-quantile range; p.u. = power unit.

Table 3-9 Statistical characteristics of the WFB training and testing dataset D.

Variables	Nº Points	Mean	Median	Std. Dev.	Skewness	Kurtosis	IQR[a]
Train Dataset							
Wind Speed (m/s)	5568	6.547	6.102	3.253	0.632	0.229	4.511
Wind power (p.u.)	5568	0.294	0.209	0.284	0.836	-0.409	0.449
Test Dataset							
Wind Speed (m/s)	4308	7.557	7.122	3.456	0.541	-0.009	4.770
Wind power (p.u.)	4308	0.321	0.262	0.263	0.524	-0.957	0.443

[a] IQR = Inter-quantile range; p.u. = power unit.

Fig. 3-109 depicts the calibration obtained for WFA using various estimators. This graph shows that NW has the best overall calibration results.

Fig. 3-110 presents sharpness results, which are better for splines QR and worse for QC. In terms of resolution, linear QR performs better than the other estimators, which are almost the same, as depicted in Fig. 3-111.

Fig. 3-112 shows that QC has better performance in terms of the skill score, while linear QR has the worst.

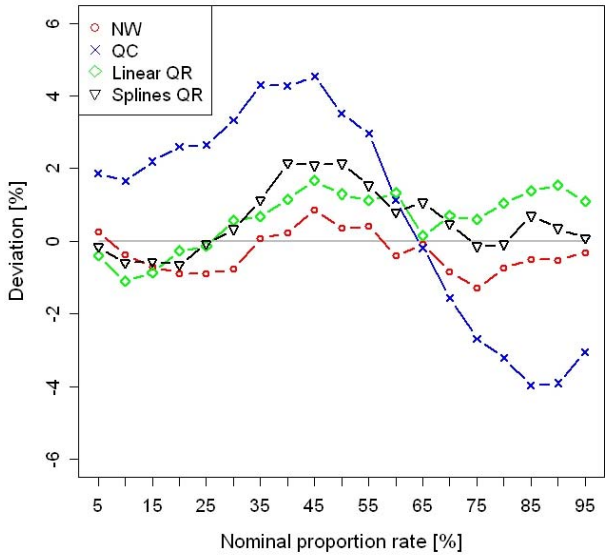

Fig. 3-109 Calibration diagram for the offline test with WFA dataset D.

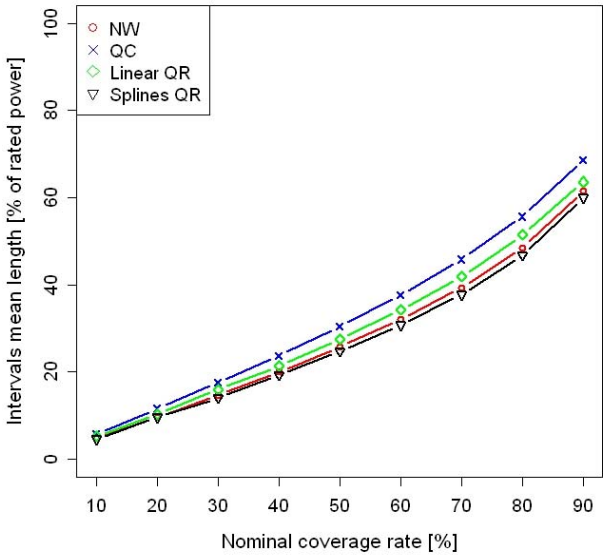

Fig. 3-110 Sharpness diagram for the offline test with WFA dataset D.

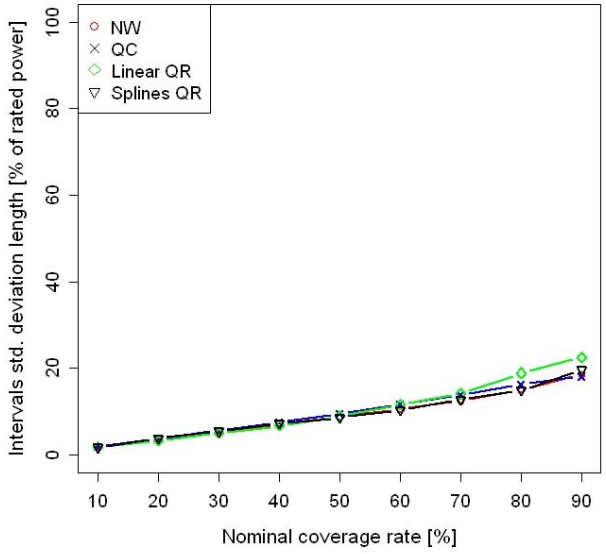

Fig. 3-111 Resolution diagram for the offline test with WFA dataset D.

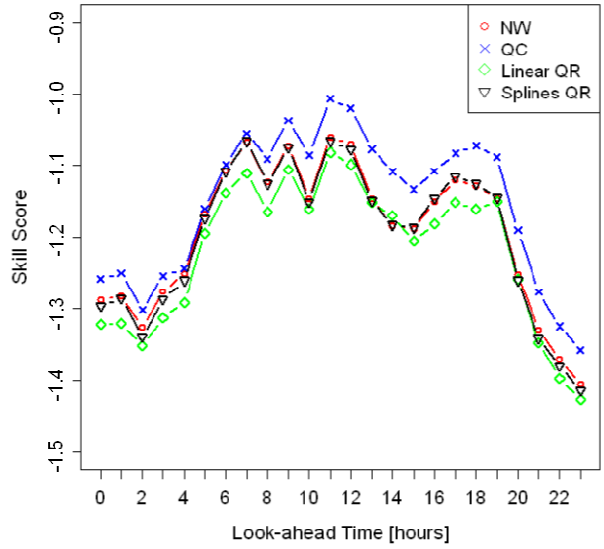

Fig. 3-112 Skill score diagram for the offline test with WFA dataset D.

Fig. 3-113 through Fig. 3-116 depict the results for WFB. The graphs show that the behavior is similar as in the WFA case, except for calibration, where for quantiles below 60%, QC has the best performance; in addition to this result, estimators tend to underestimate quantiles.

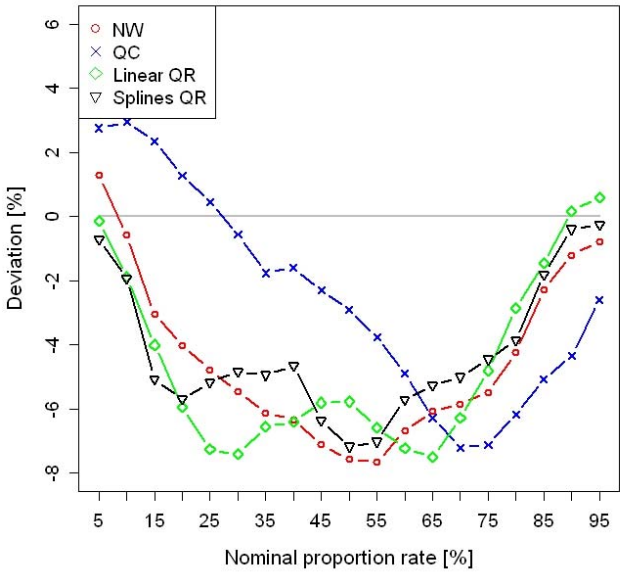

Fig. 3-113 Calibration diagram for the offline test with WFB dataset D.

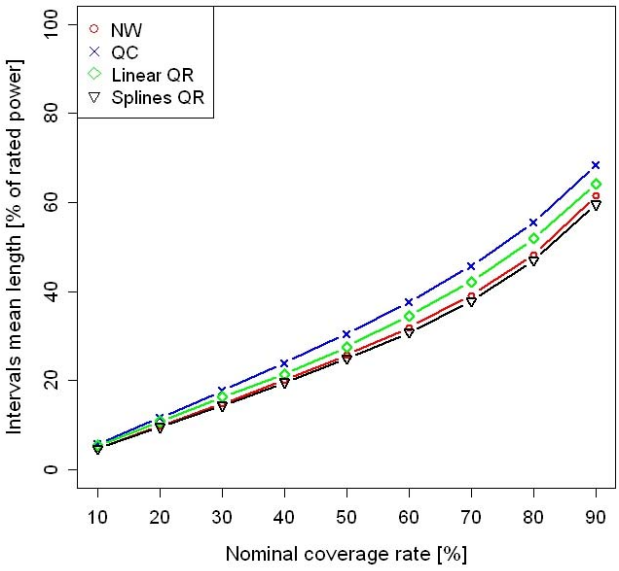

Fig. 3-114 Sharpness diagram for the offline test with WFB dataset D.

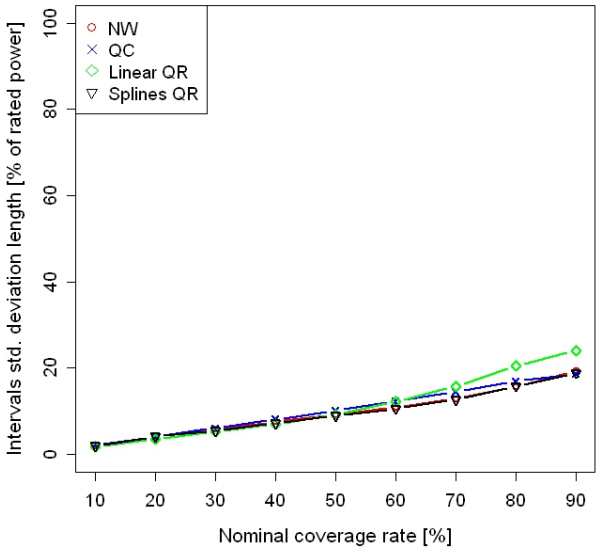

Fig. 3-115 Resolution diagram for the offline test with WFB dataset D.

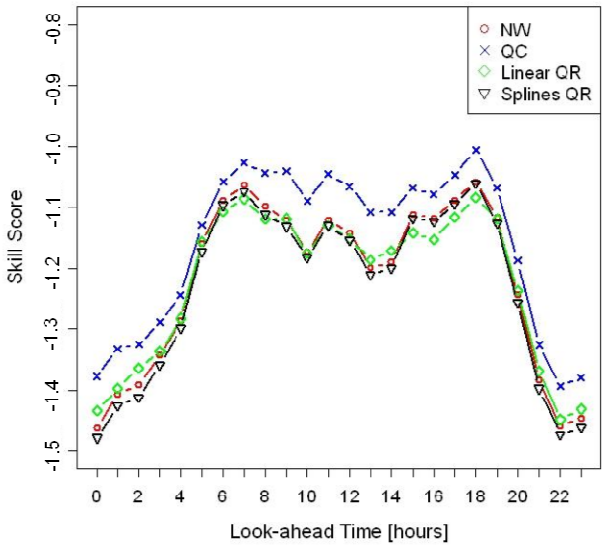

Fig. 3-116 Skill score diagram for the offline test with WFB dataset D.

The main conclusions for dataset D are that KDF estimators have the best calibration for quantiles below 60% (NW in WFA and QC in WFB); splines QR has the best sharpness and linear QR the best resolution; and in terms of the skill score, QC has the best performance.

Dataset E

The main characteristics of the training and testing datasets E of WFA are presented in Table 3-10, and for WFB are presented in Table 3-11. The training period was between January 2, 2009, and August 31, 2009; and the testing period was from September 1, 2009, until February 20, 2010.

The statistical field that changes most in this dataset is kurtosis. In WFA, wind speed kurtosis is positive and increases 4 times, while in WFB, it increases almost 5 times. Power kurtoses are negative. Moreover, the training period is longer than the testing one in both wind farms.

Table 3-10 Statistical characteristics of the WFA training and testing dataset E.

Variables	Nº Points	Mean	Median	Std. Dev.	Skewness	Kurtosis	IQR*
Train Dataset							
Wind Speed (m/s)	5736	7.338	6.912	3.360	0.551	0.040	4.483
Wind power (p.u.)	5636	0.319	0.262	0.270	0.516	-0.964	0.459
Test Dataset							
Wind Speed (m/s)	3936	7.104	6.808	3.305	0.557	0.164	4.527
Wind power (p.u.)	3936	0.327	0.244	0.300	0.696	-0.713	0.483

[a] IQR = Inter-quantile range; p.u. = power unit.

Table 3-11 Statistical characteristics of the WFB training and testing dataset E.

Variables	Nº Points	Mean	Median	Std. Dev.	Skewness	Kurtosis	IQR
Train Dataset							
Wind Speed (m/s)	5796	7.147	6.735	3.402	0.574	0.053	4.667
Wind power (p.u.)	5796	0.296	0.229	0.258	0.624	-0.809	0.435
Test Dataset							
Wind Speed (m/s)	4080	6.761	6.384	3.336	0.629	0.234	4.602
Wind power (p.u.)	4080	0.319	0.237	0.297	0.740	-0.645	0.474

[a] IQR = Inter-quantile range; p.u. = power unit.

Fig. 3-117 depicts the calibration obtained for WFA using various estimators. This graph shows that QR approaches have better overall calibration results, although KDF estimators become better performers for quantiles above 65%.

Fig. 3-118 presents sharpness results, which are better for splines QR and worse for QC. In terms of resolution, all estimators have similar results (Fig. 3-119). The splines QR has better performance in terms of the skill score, while linear QR has the worst. Hence, the skill score performance of KDF estimators lies between the QR approaches, with NW performing better than QC, as shown in Fig. 3-120.

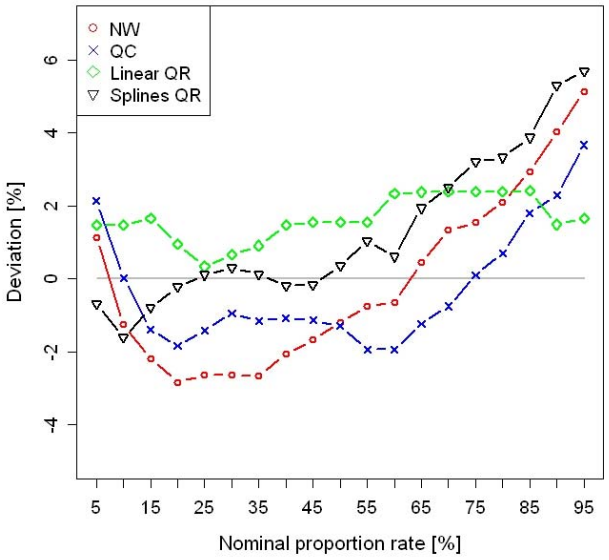

Fig. 3-117 Calibration diagram for the offline test with WFA dataset E.

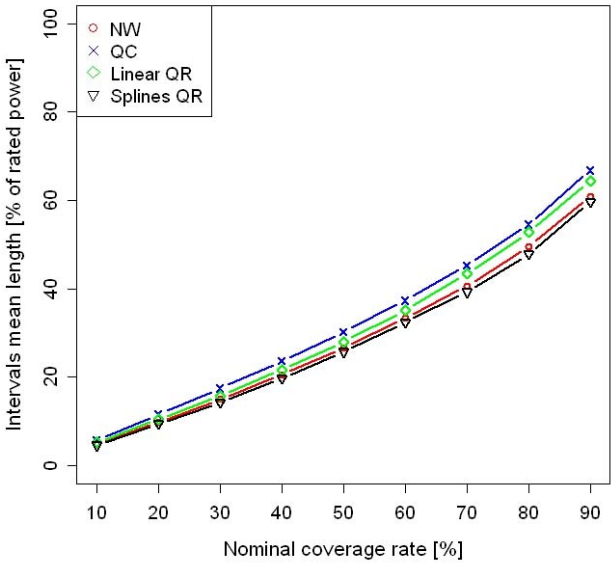

Fig. 3-118 Sharpness diagram for the offline test with WFA dataset E.

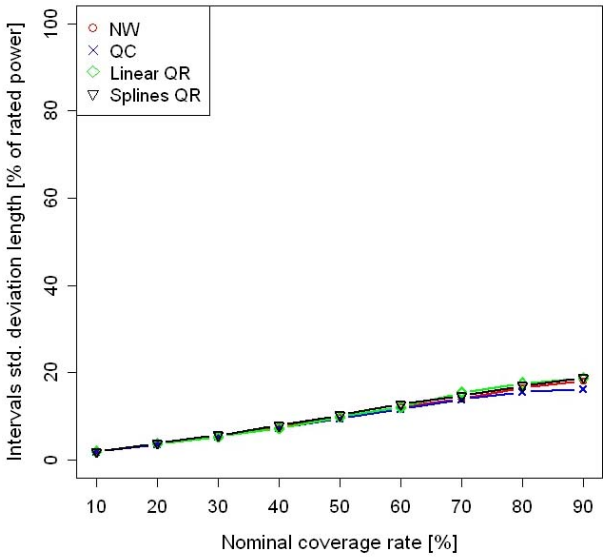

Fig. 3-119 Resolution diagram for the offline test with WFA dataset E.

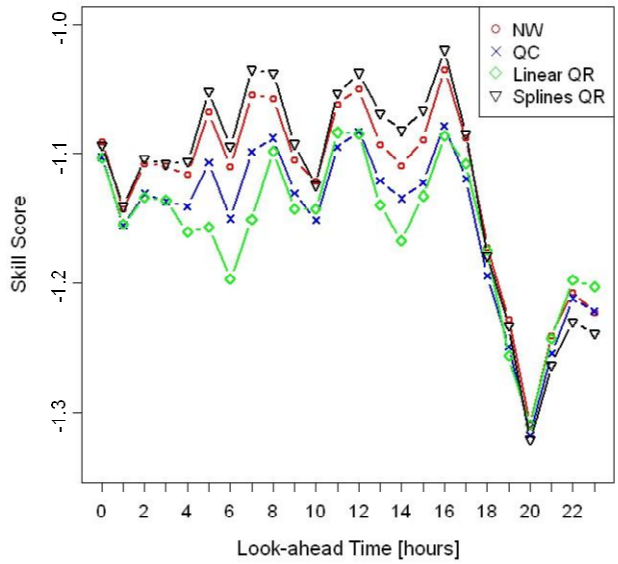

Fig. 3-120 Skill score diagram for the offline test with WFA dataset E.

Fig. 3-121 through Fig. 3-124 depict the results for WFB. The graphs show that the behavior is similar as in the WFA case, except for the calibration and resolution. In fact, for quantiles above 25%, the KDF estimators become better performers than splines QR, where linear QR has the worst calibration performance; moreover, quantiles tend to be overestimated. As for resolution, linear QR performs worse than do the other estimators, where results are generally the same.

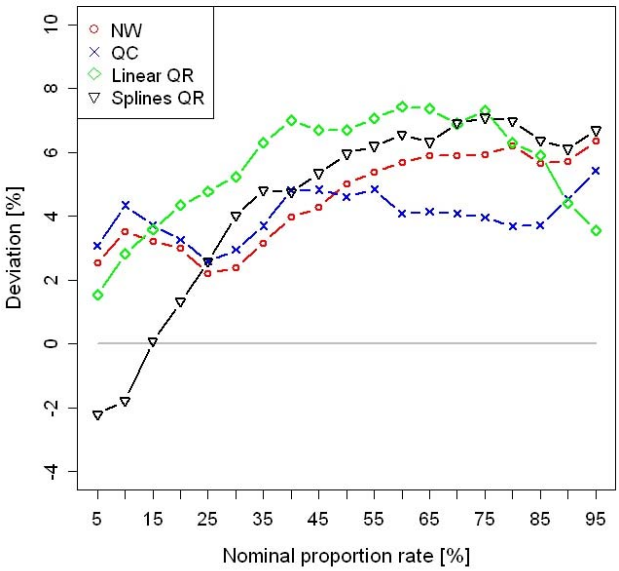

Fig. 3-121 Calibration diagram for the offline test with WFB dataset E.

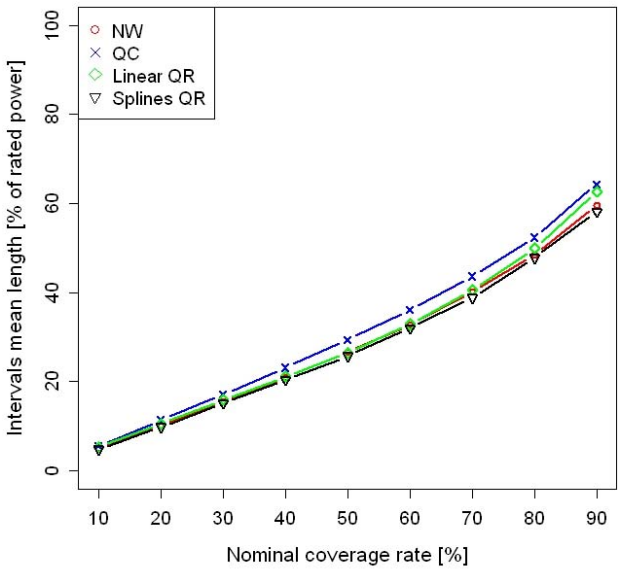

Fig. 3-122 Sharpness diagram for the offline test with WFB dataset E.

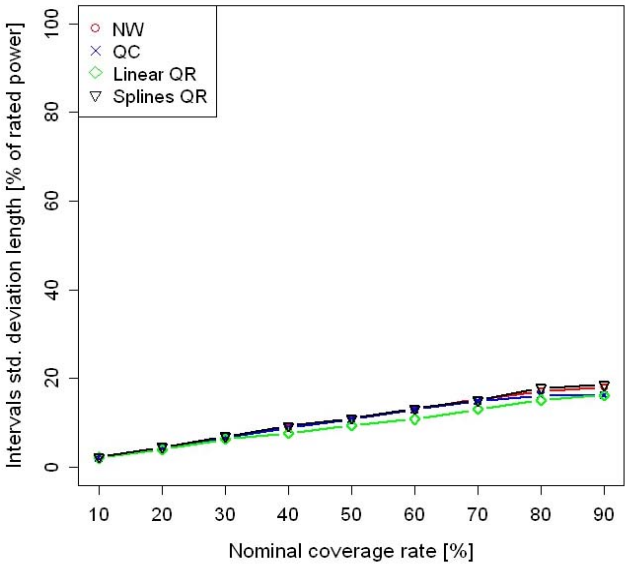

Fig. 3-123 Resolution diagram for the offline test with WFB dataset E.

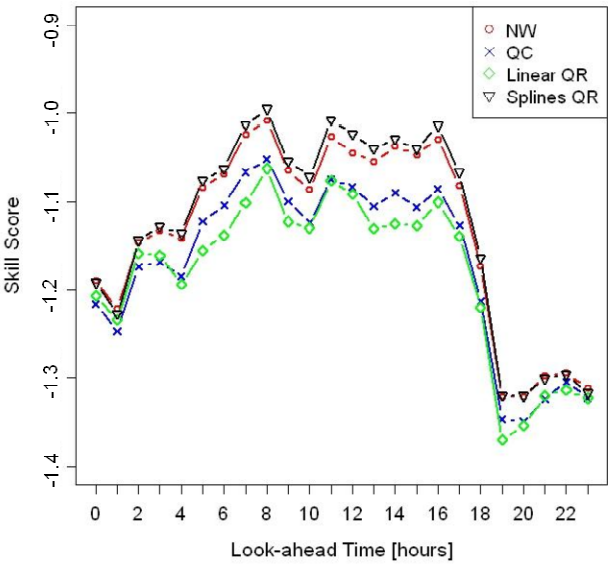

Fig. 3-124 Skill score diagram for the offline test with WFB dataset E.

The main conclusions for dataset E are that QR estimators have better overall calibration, although KDF approaches also present good performance, particularly in WFB; splines QR has the best sharpness and linear QR the worst resolution; and in terms of the skill score, QR approaches have the best and the worst performance and, among the KDF estimators, NW is better.

Dataset F

The main characteristics of the training and testing datasets F of WFA are presented in Table 3-12, and for WFB are presented in Table 3-13. The training period was from September 1, 2009, until February 20, 2010; and the testing period was between January 2, 2009, and August 31, 2009.

The statistical field that changes the most in this dataset is kurtosis. In WFA, wind speed kurtosis is positive and decreases 4 times, while in WFB, it decreases almost 5 times. Power kurtoses are negative. Moreover, the training period is shorter than the testing period for both wind farms.

Table 3-12 Statistical characteristics of the WFA training and testing dataset F.

Variables	Nº Points	Mean	Median	Std. Dev.	Skewness	Kurtosis	IQR[a]
Train Dataset							
Wind Speed (m/s)	3936	7.104	6.808	3.305	0.557	0.164	4.527
Wind power (p.u.)	3936	0.327	0.244	0.300	0.696	-0.713	0.483
Test Dataset							
Wind Speed (m/s)	5736	7.338	6.912	3.360	0.551	0.040	4.483
Wind power (p.u.)	5636	0.319	0.262	0.270	0.516	-0.964	0.459

[a] IQR = Inter-quantile range; p.u. = power unit.

Table 3-13 Statistical characteristics of the WFB training and testing dataset F.

Variables	Nº Points	Mean	Median	Std. Dev.	Skewness	Kurtosis	IQR[a]
Train Dataset							
Wind Speed (m/s)	4080	6.761	6.384	3.336	0.629	0.234	4.602
Wind power (p.u.)	4080	0.319	0.237	0.297	0.740	-0.645	0.474
Test Dataset							
Wind Speed (m/s)	5796	7.147	6.735	3.402	0.574	0.053	4.667
Wind power (p.u.)	5796	0.296	0.229	0.258	0.624	-0.809	0.435

[a] IQR = Inter-quantile range; p.u. = power unit.

Fig. 3-125 depicts the calibration obtained for WFA using various estimators. This graph shows that for quantiles below 50%, QC has the best calibration results, and above 50%, splines QR becomes the better performer. These approaches underestimate quantiles.

Fig. 3-126 presents sharpness results, which are better for splines QR and worse for QC. In terms of resolution, for quantiles below (above) 70%, linear QR performs worse (better) than the other estimators, which are almost the same, as depicted in Fig. 3-127.

Fig. 3-128 shows that QC has better performance in terms of the skill score, while linear QR has the worst.

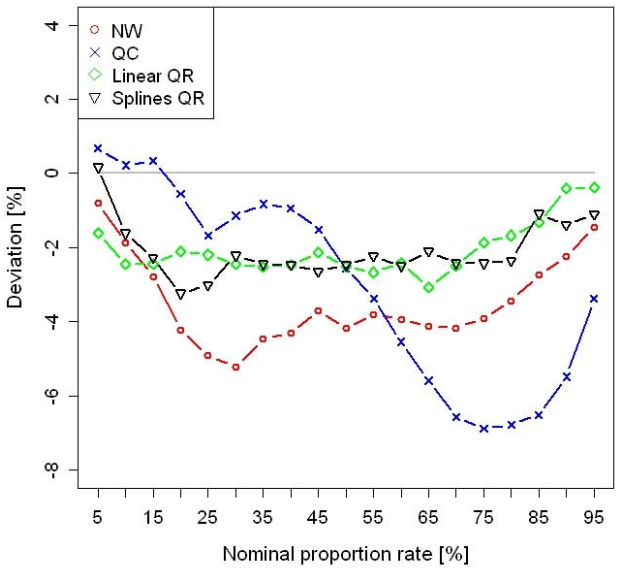

Fig. 3-125 Calibration diagram for the offline test with WFA dataset F.

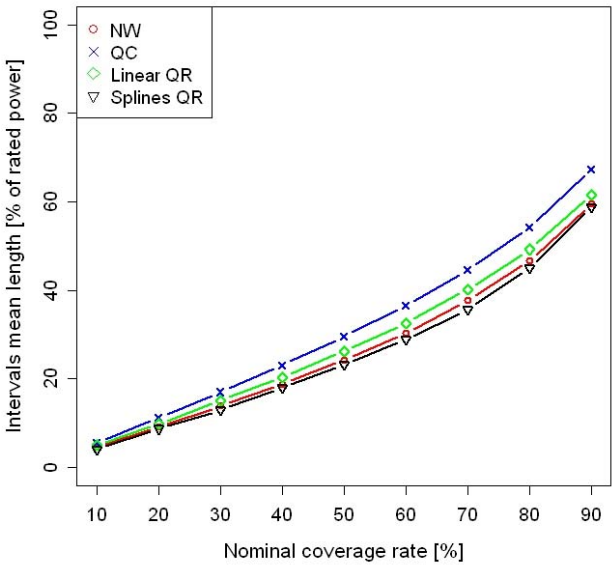

Fig. 3-126 Sharpness diagram for the offline test with WFA dataset F.

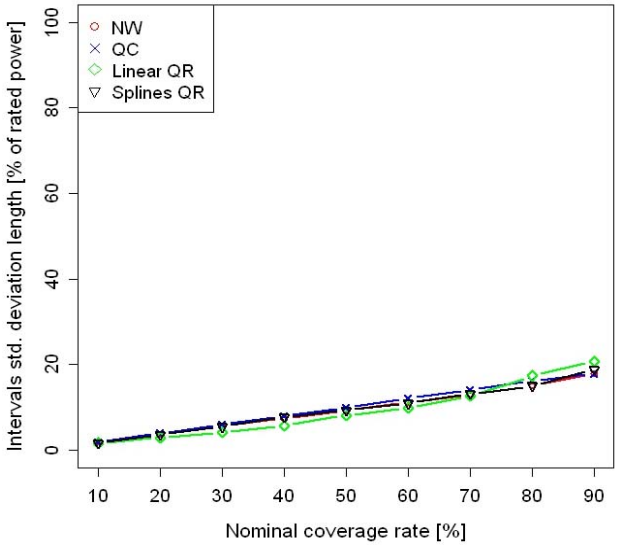

Fig. 3-127 Resolution diagram for the offline test with WFA dataset F.

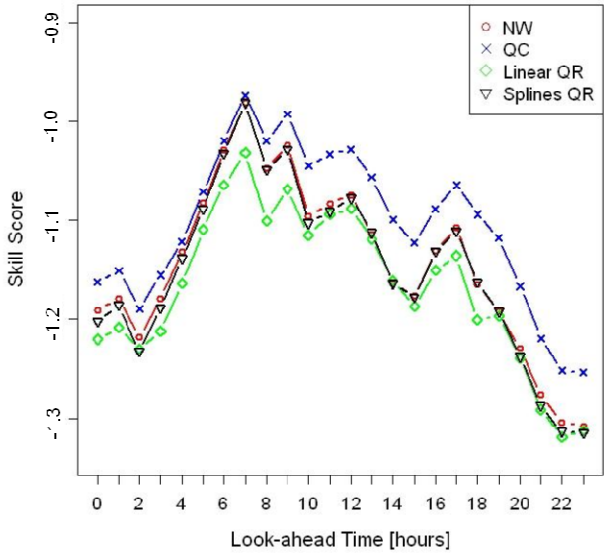

Fig. 3-128 Skill score diagram for the offline test with WFA dataset F.

Fig. 3-129 through Fig. 3-132 depict the results for WFB. The graphs show that the behavior is similar as in the WFA case.

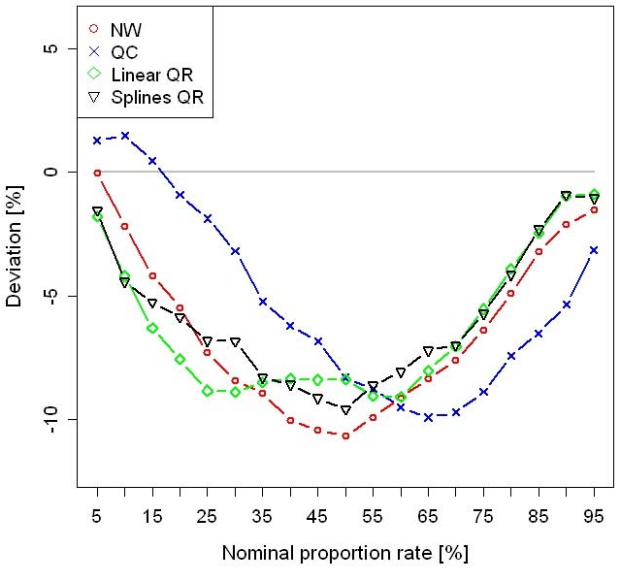

Fig. 3-129 Calibration diagram for the offline test with WFB dataset F.

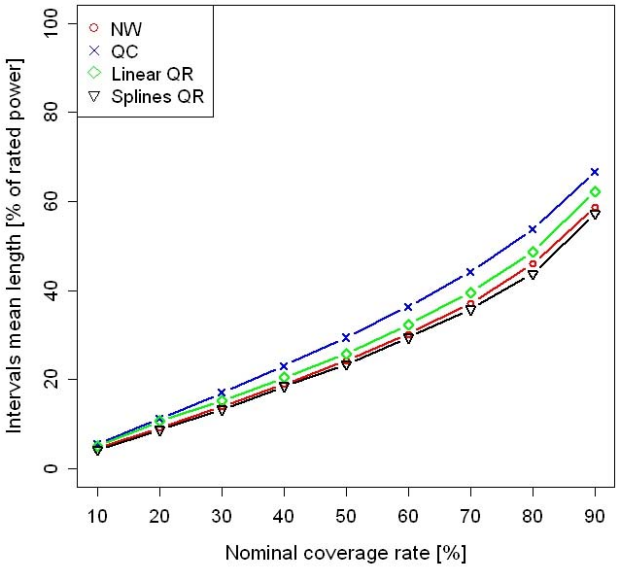

Fig. 3-130 Sharpness diagram for the offline test with WFB dataset F.

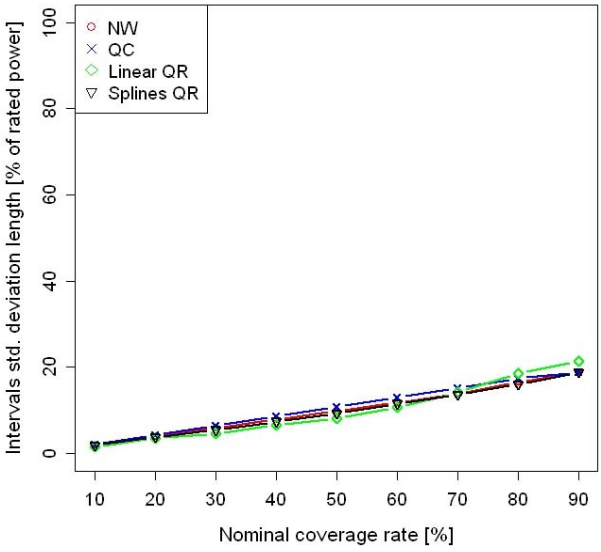

Fig. 3-131 Resolution diagram for the offline test with WFB dataset F.

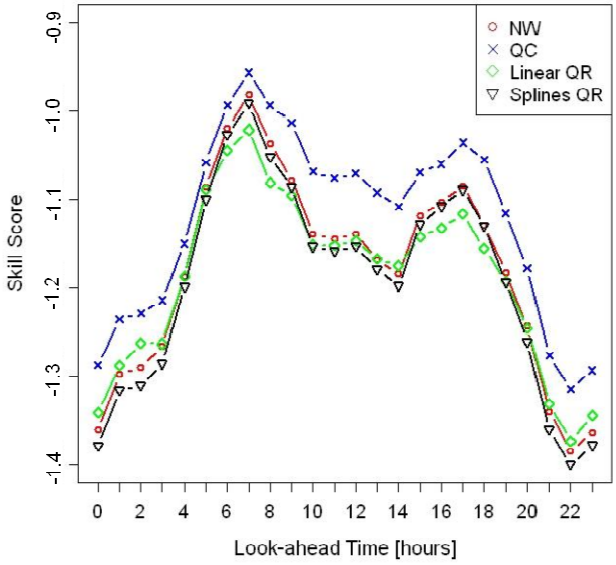

Fig. 3-132 Skill score diagram for the offline test with WFB dataset F.

The main conclusions for dataset F are that QC has the best calibration for quantiles below 60%; splines QR has the best sharpness and linear QR the best resolution for quantiles above 70%; and in terms of the skill score, QC has the best performance.

Dataset Results

In spite of similar results for both wind farms, outputs were revealed to change under different data characteristics between training and testing. In fact, if wind speed and power kurtosis increase (in module) from training to testing, regardless of their sign and whether the training dataset is larger or smaller than the testing dataset, QR estimators have the best calibration, although for WFB, KDF approaches are better for quantiles above 30%. QR approaches have the best sharpness and the worst resolution, namely, splines in the former and linear QR in the latter. As for the skill score, QR estimators deliver the best and the worst performances, and NW is better than QC.

If, on the other hand, wind and power kurtosis decrease (in module) from training to testing, regardless their sign and whether the training dataset is larger or smaller than the testing dataset, QC is better for quantiles below 60%, although in dataset D, NW has the better performance instead. QR approaches have the best sharpness and resolution. In terms of the skill score, QC is the estimator with the best results.

3.4.3.4 Evaluation with Different Parameters

In order to assess the performance of the estimators under different parameters, several tests were run for each kernel size combination, using the *Chen 1* kernel with dataset A, for both wind farms. Results are as follows. The kernel size values were determined experimentally (via trial and error) and use as a starting point the values suggested by the function *cde.bandwidths* from the R package "hdrcde" [58].

Kernel size: (h_{Power}; $h_{WindSpeed}$) = (0.002; 0.04)

Fig. 3-133 depicts the calibration obtained for WFA using various estimators. This graph shows that QR approaches have better calibration results. KDF estimators tend to underestimate the quantiles.

Fig. 3-134 presents sharpness and Fig. 3-135 resolution. In both, splines QR has the best performance and QC the worst, although resolution results are very similar among estimators.

The splines QR has better performance in terms of the skill score, whereas linear QR has the worst. Hence, the skill score performance of KDF estimators lies between the QR approaches, with NW performing better than QC, as shown in Fig. 3-136.

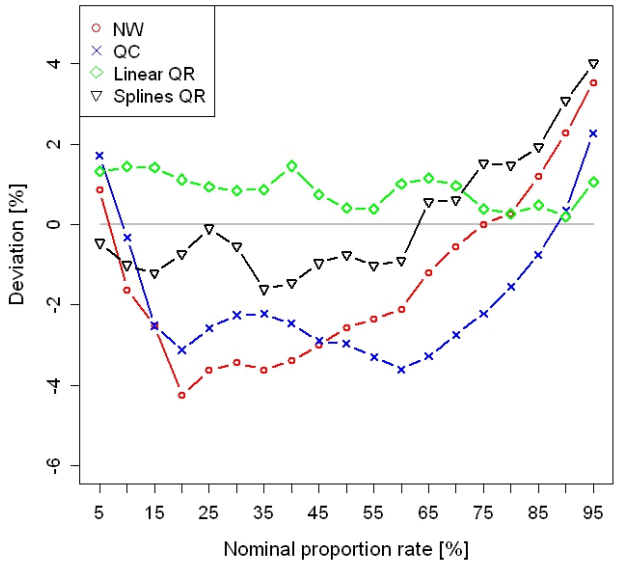

Fig. 3-133 Calibration diagram for the offline test with WFA dataset A.

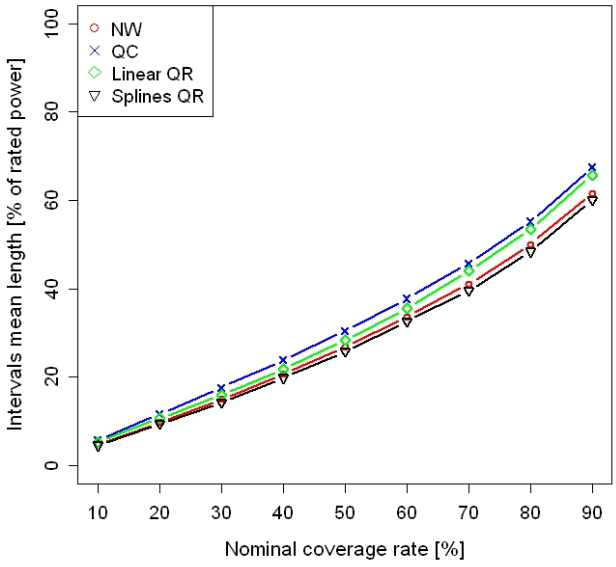

Fig. 3-134 Sharpness diagram for the offline test with WFA dataset A.

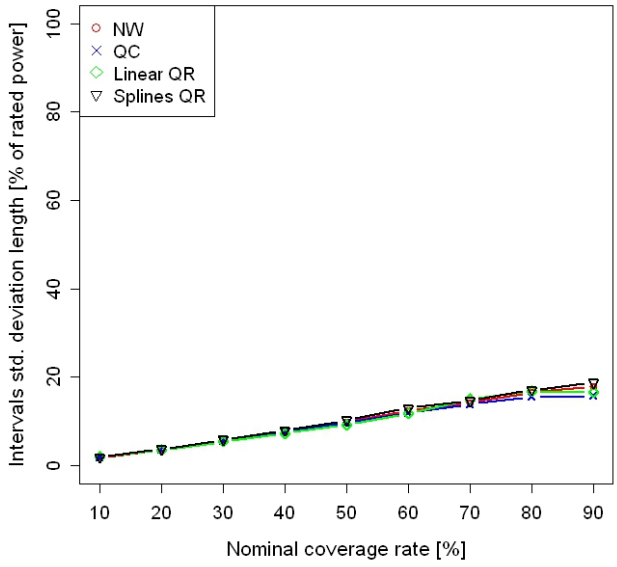

Fig. 3-135 Resolution diagram for the offline test with WFA dataset A.

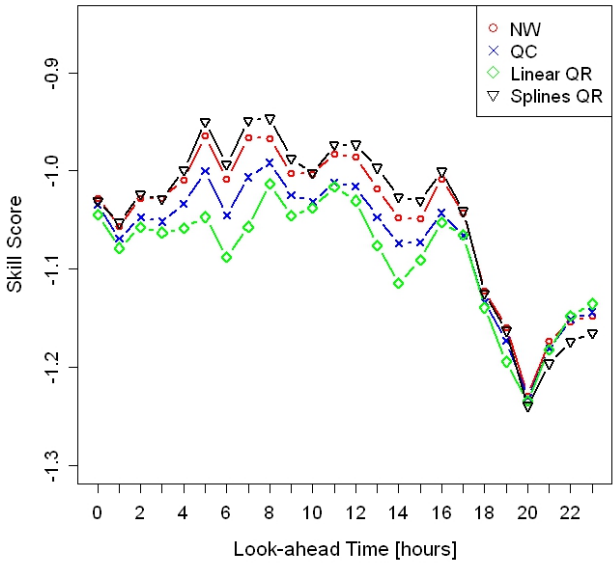

Fig. 3-136 Skill score diagram for the offline test with WFA dataset A.

Fig. 3-137 through Fig. 3-140 depict the results for WFB. The graphs show that the behavior is similar as in the WFA case, except for the calibration and resolution, in which linear QR is clearly the worst (see Fig. 3-139). For quantiles above 25%, the KDF estimators become better performers than splines QR, with linear QR being the approach with the worst calibration performance. Moreover, the estimators tend to overestimate the quantiles.

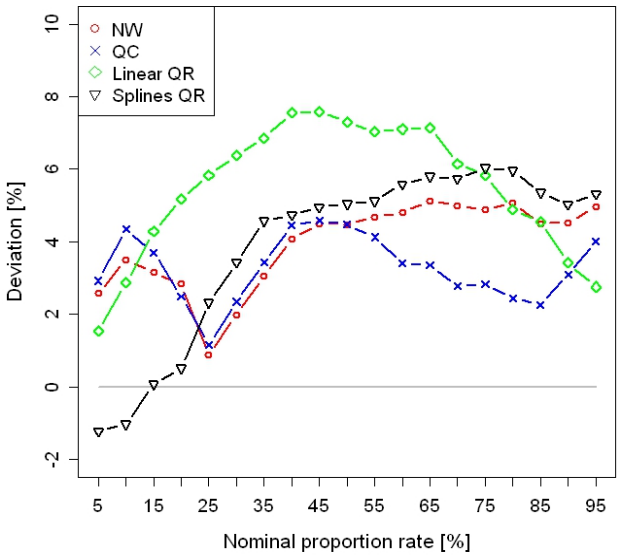

Fig. 3-137 Calibration diagram for the offline test with WFB dataset A.

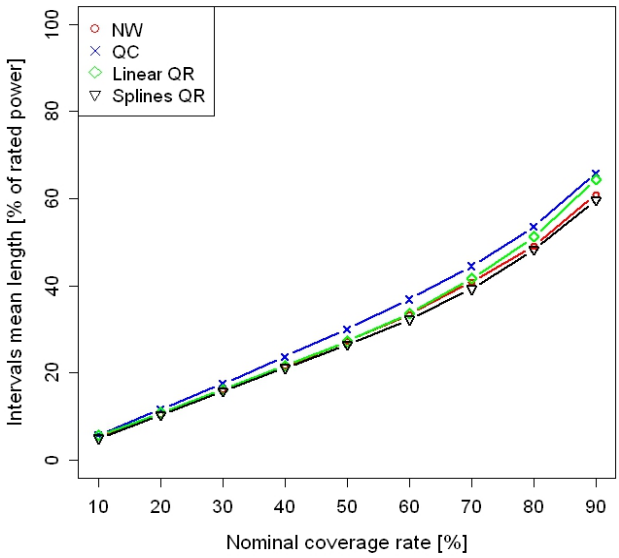

Fig. 3-138 Sharpness diagram for the offline test with WFB dataset A.

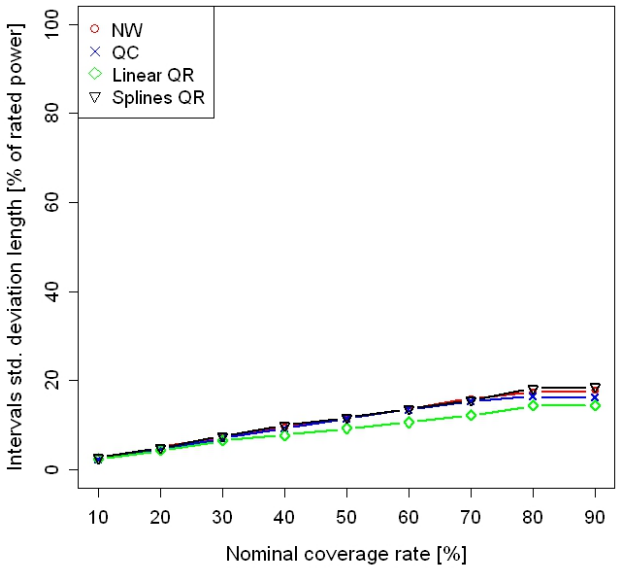

Fig. 3-139 Resolution diagram for the offline test with WFB dataset A.

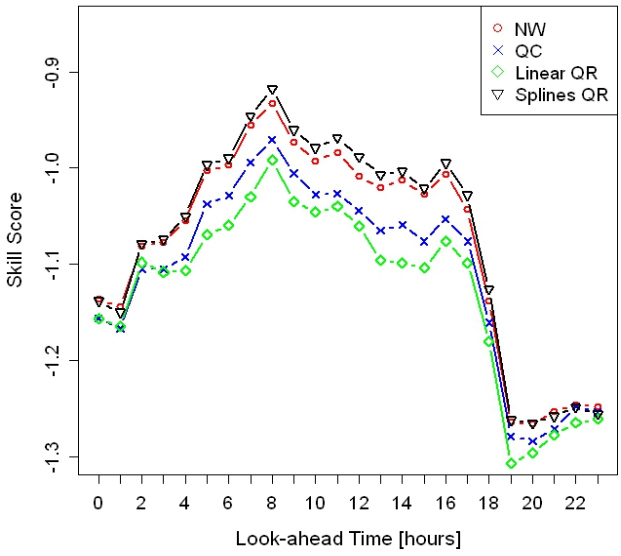

Fig. 3-140 Skill score diagram for the offline test with WFB dataset A.

The main conclusions for this kernel size choice are that splines QR has the best overall calibration in WFA, although in WFB, KDF approaches in particular have a better performance; splines QR has the best sharpness and resolution; and in terms of the skill score, QR approaches have the best and the worst performance (splines QR and linear QR, respectively) and, among the KDF estimators, NW is better.

Kernel size: ($h_{Power;}$ $h_{WindSpeed}$) = (0.004; 0.02)

Fig. 3-141 depicts the calibration obtained for WFA using various estimators. This graph shows that QR approaches have better calibration results. Estimators tend to underestimate the quantiles.

Fig. 3-142 presents sharpness and Fig. 3-143 resolution. In both, splines QR has the best performance and linear QR the worst, although resolution results are very similar among estimators.

The splines QR has better performance in terms of the skill score, while linear QR has the worst. Hence, the skill score performance of KDF estimators lies between the QR approaches, with NW performing better than QC, as shown in Fig. 3-144.

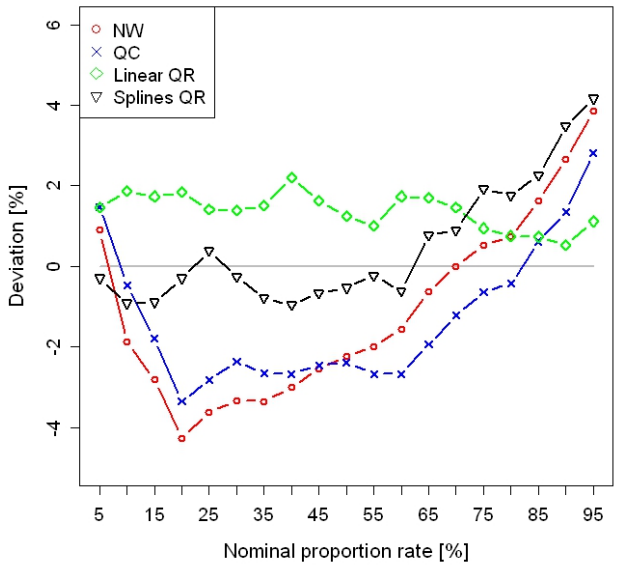

Fig. 3-141 Calibration diagram for the offline test with WFA dataset A.

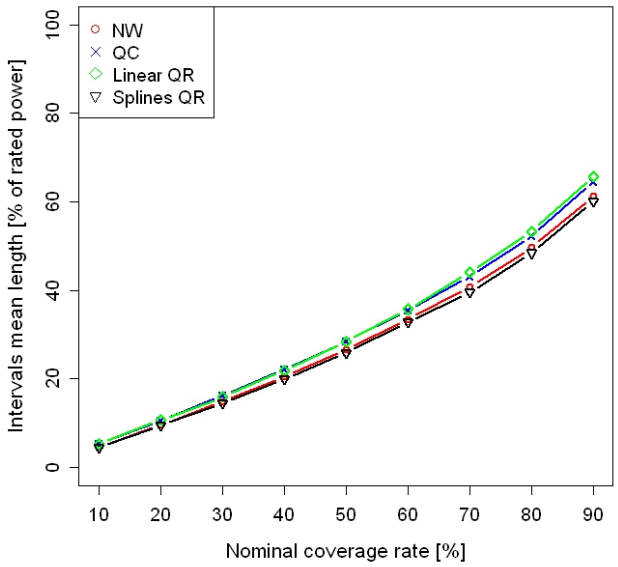

Fig. 3-142 Sharpness diagram for the offline test with WFA dataset A.

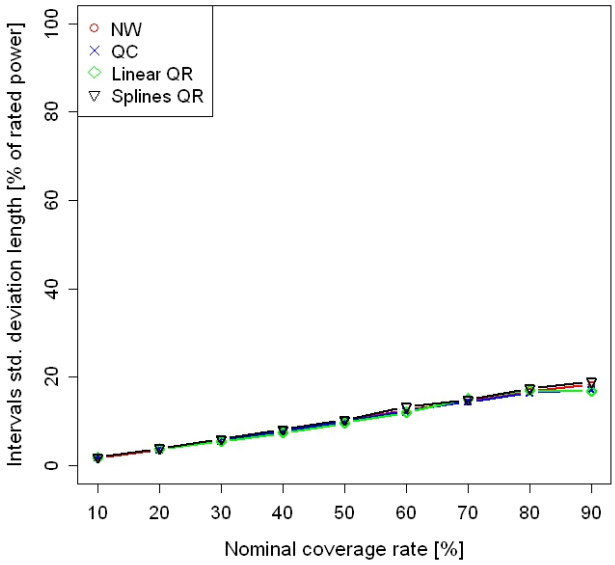

Fig. 3-143 Resolution diagram for the offline test with WFA dataset A.

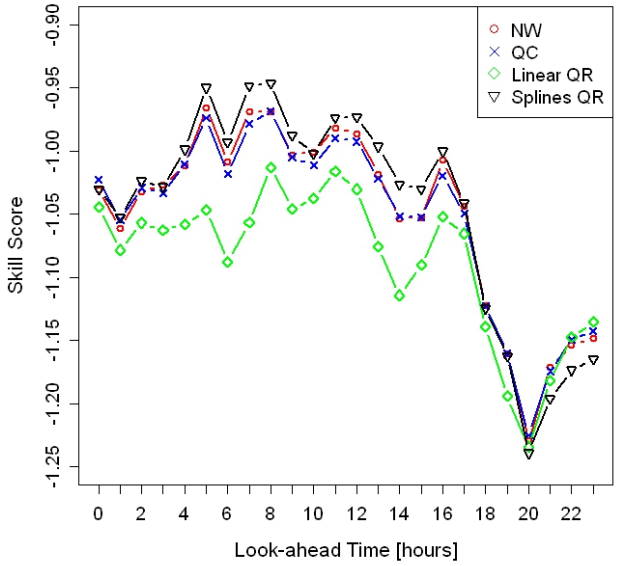

Fig. 3-144 Skill score diagram for the offline test with WFA dataset A.

Fig. 3-145 through Fig. 3-148 depict the results for WFB. It is shown that the behavior is similar as in the WFA case, except for the calibration. In fact, for quantiles above 25%, the KDF estimators become better performers than splines QR, with linear QR being the approach with the worst calibration performance. Moreover, quantiles tend to be overestimated.

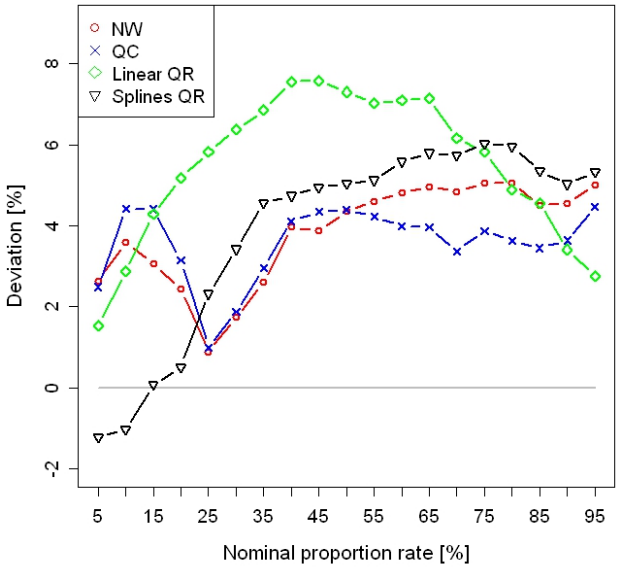

Fig. 3-145 Calibration diagram for the offline test with WFB dataset A.

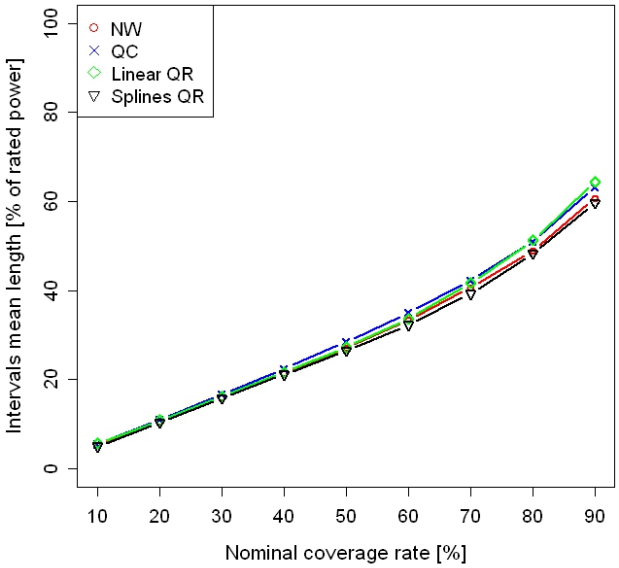

Fig. 3-146 Sharpness diagram for the offline test with WFB dataset A.

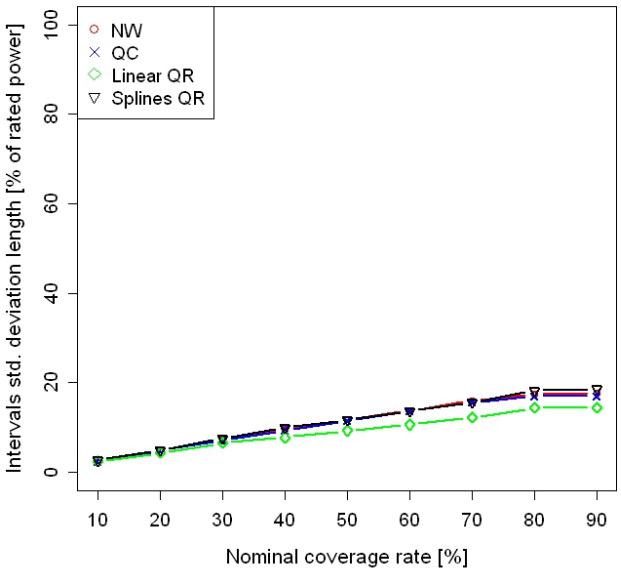

Fig. 3-147 Resolution diagram for the offline test with WFB dataset A.

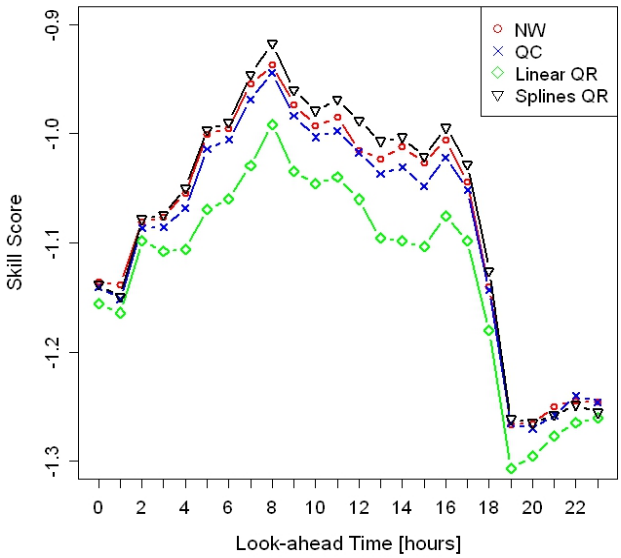

Fig. 3-148 Skill score diagram for the offline test with WFB dataset A.

The main conclusions for this kernel size choice are that splines QR has the best overall calibration in WFA, although in WFB, KDF approaches are better; splines QR has the best sharpness and resolution; and in terms of the skill score QR approaches have the best and the worst performance and, among the KDF estimators, NW is better.

Kernel size: $(h_{Power}; h_{WindSpeed}) = (0.01; 0.05)$

Fig. 3-149 depicts the calibration obtained for WFA using various estimators. This graph shows that QR approaches are better than KDF estimators, which tend to underestimate the quantiles.

Fig. 3-150 presents sharpness and Fig. 3-151 resolution. In both, splines QR has the best performance and QC the worst, although resolution results are very similar among estimators.

The splines QR has better performance in terms of the skill score, while QC and linear QR are worse, as shown in Fig. 3-152. Among the KDF estimators NW is the best.

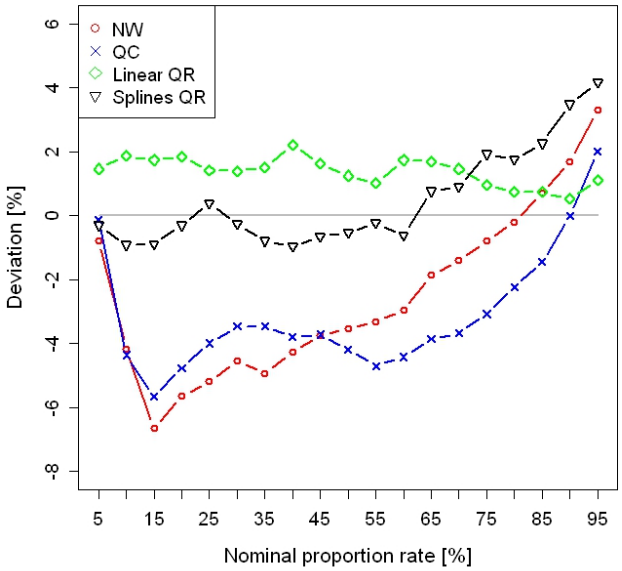

Fig. 3-149 Calibration diagram for the offline test with WFA dataset A.

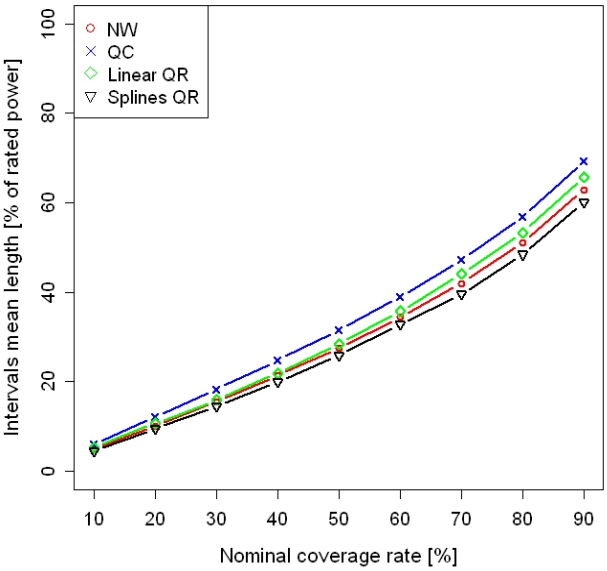

Fig. 3-150 Sharpness diagram for the offline test with WFA dataset A.

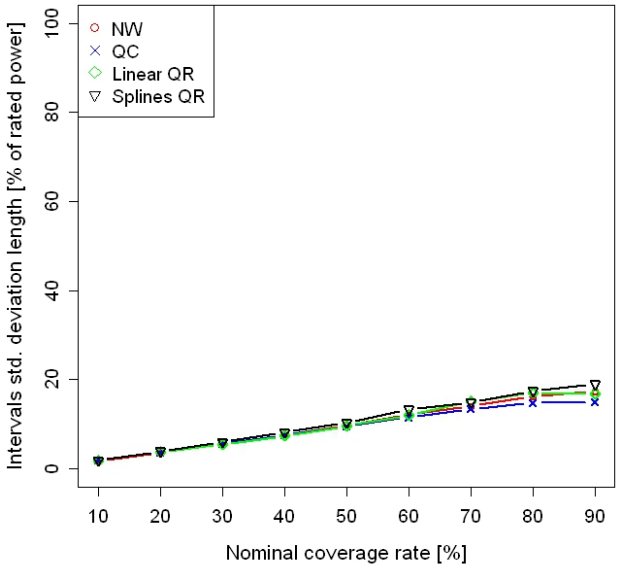

Fig. 3-151 Resolution diagram for the offline test with WFA dataset A.

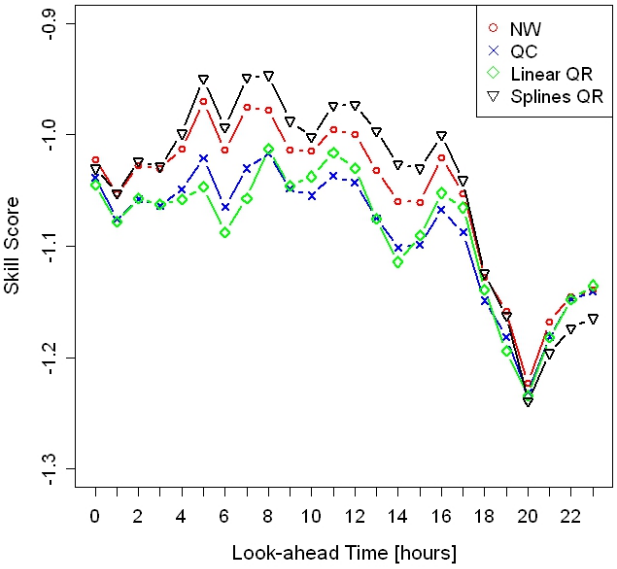

Fig. 3-152 Skill score diagram for the offline test with WFA dataset A.

Fig. 3-153 through Fig. 3-156 depict the results for WFB. The graphs show that the behavior is similar as in the WFA case, except for the resolution, where linear QR is the worst, and calibration, where KDF estimators have better overall calibration than splines QR, with linear QR being the approach with the worst calibration performance and QC the best.

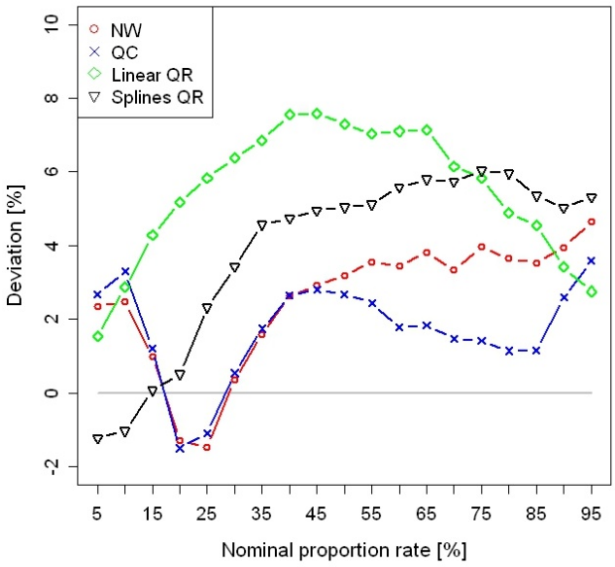

Fig. 3-153 Calibration diagram for the offline test with WFB dataset A.

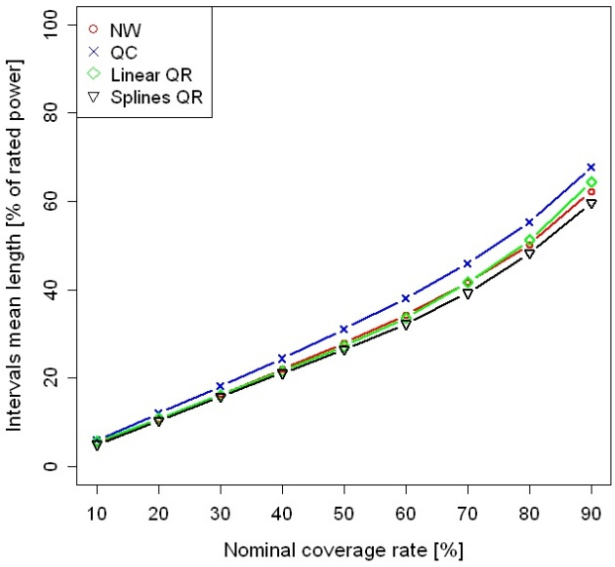

Fig. 3-154 Sharpness diagram for the offline test with WFB dataset A.

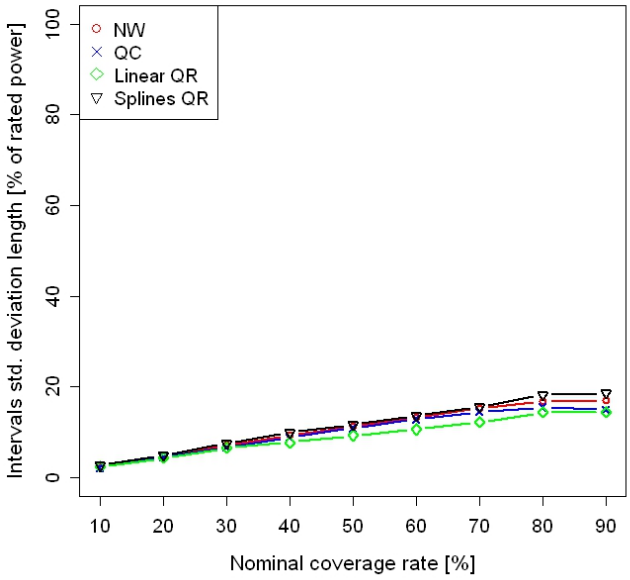

Fig. 3-155 Resolution diagram for the offline test with WFB dataset A.

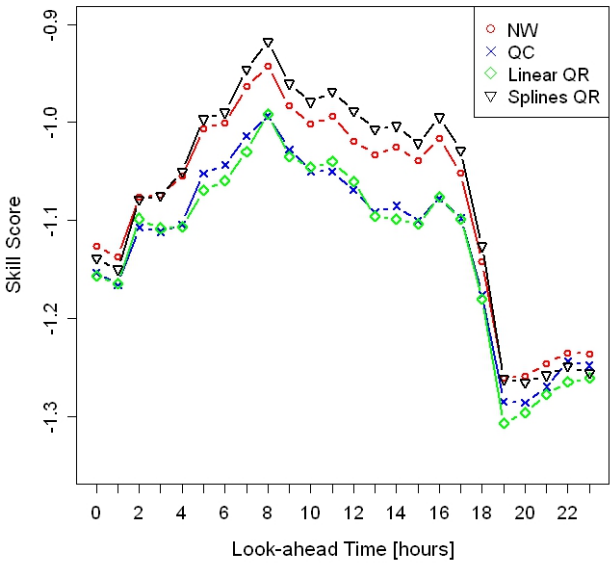

Fig. 3-156 Skill score diagram for the offline test with WFB dataset A.

The main conclusions for this kernel size choice are that splines QR has the best overall calibration in WFA, although KDF approaches perform better in WFB; splines QR has the best sharpness and resolution; and in terms of the skill score, splines QR has the best performance and, among the KDF estimators, NW is better.

Kernel size: ($h_{Power;}$ $h_{WindSpeed}$) = (0.004; 0.04)

Fig. 3-157 depicts the calibration obtained for WFA using various estimators. This graph shows that QR approaches have better calibration results and that KDF estimators tend to underestimate the quantiles, although their performance is better than in the previous kernel choice.

Fig. 3-158 presents sharpness and Fig. 3-159 resolution. In both, splines QR has the best performance and QC the worst.

The splines QR has better performance in terms of the skill score, while linear QR has the worst, as shown in Fig. 3-160. Among the KDF approaches, NW is better.

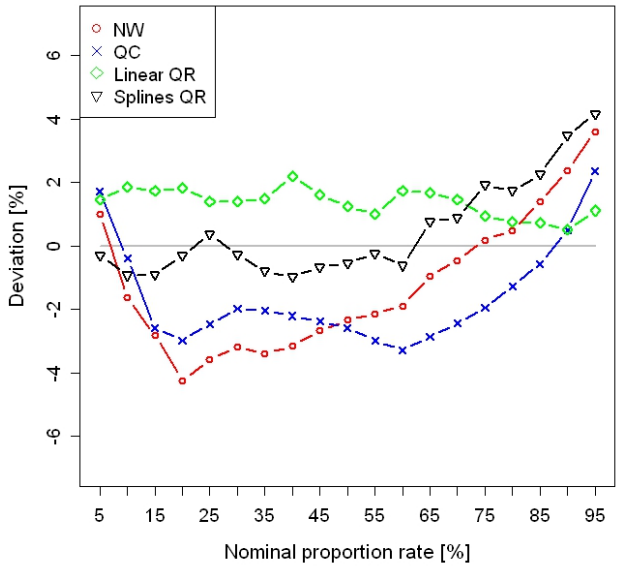

Fig. 3-157 Calibration diagram for the offline test with WFA dataset A.

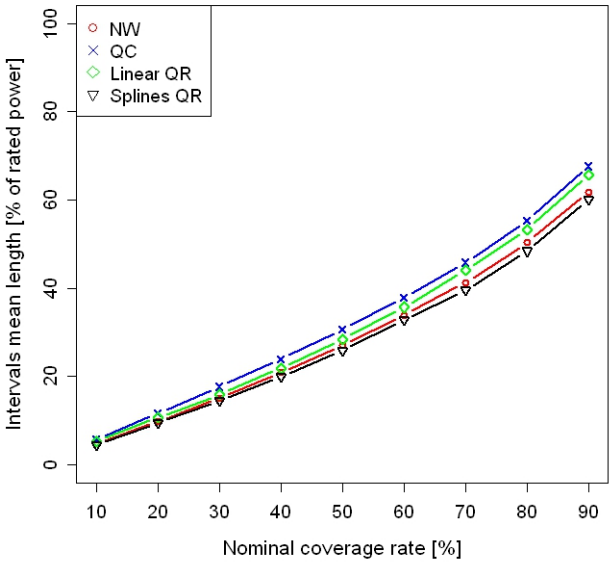

Fig. 3-158 Sharpness diagram for the offline test with WFA dataset A.

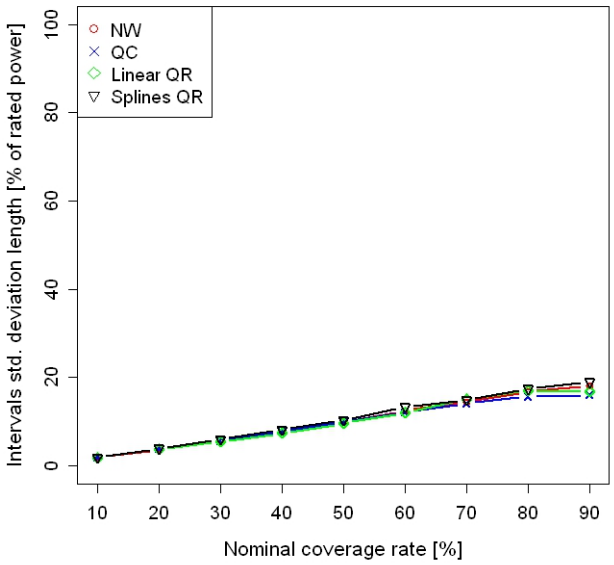

Fig. 3-159 Resolution diagram for the offline test with WFA dataset A.

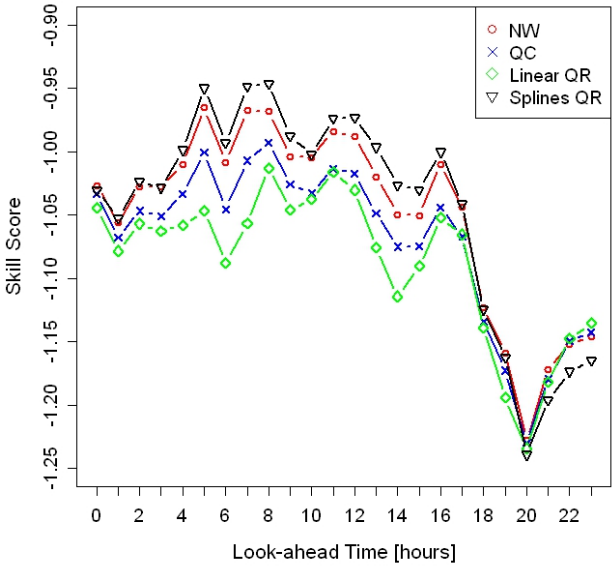

Fig. 3-160 Skill score diagram for the offline test with WFA dataset A.

Fig. 3-161 through Fig. 3-164 depict the results for WFB. The graphs show that the behavior is similar as in the WFA case, except for the resolution, in which linear QR is the worst, and calibration, where KDF estimators have better overall calibration than splines QR, with linear QR being the approach with the worst performance and NW the best.

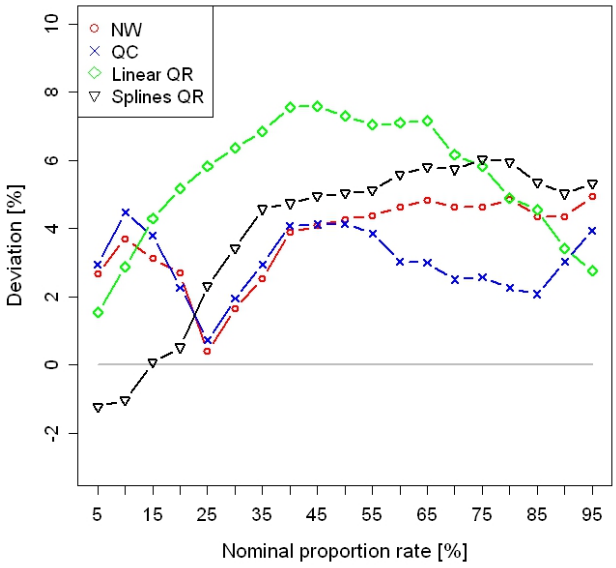

Fig. 3-161 Calibration diagram for the offline test with WFB dataset A.

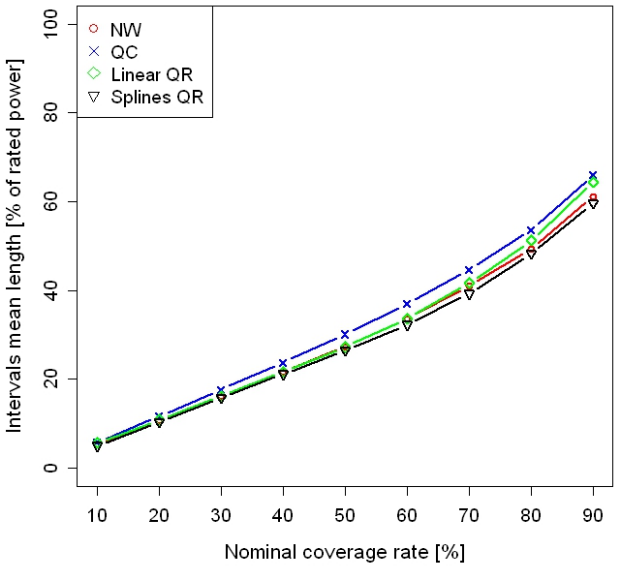

Fig. 3-162 Sharpness diagram for the offline test with WFB dataset A.

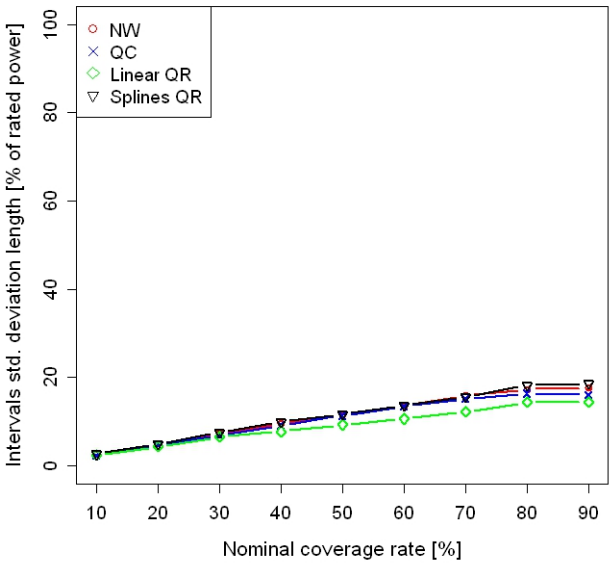

Fig. 3-163 Resolution diagram for the offline test with WFB dataset A.

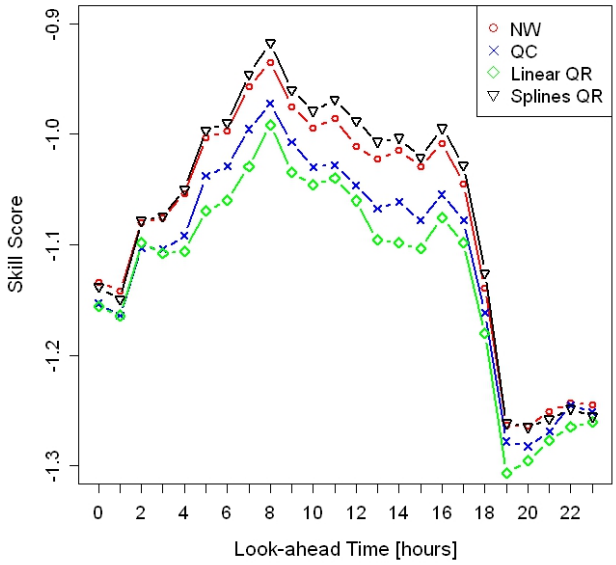

Fig. 3-164 Skill score diagram for the offline test with WFB dataset A.

The main conclusions for this kernel size choice are that splines QR has the best overall calibration in WFA, although KDF approaches perform better in WFB; splines QR has the best sharpness and resolution; and in terms of the skill score, splines QR has the best performance and, among the KDF approaches NW is better.

Kernel size: $(h_{Power;}\ h_{WindSpeed}) = (0.008;\ 0.05)$

Fig. 3-165 depicts the calibration obtained for WFA using various estimators. This graph shows that QR approaches have better calibration results and that KDF estimators tend to underestimate the quantiles.

Fig. 3-166 presents sharpness and Fig. 3-167 resolution. In both, splines QR has the best performance and QC the worst, although resolution results are very similar among estimators.

The splines QR has better performance in terms of skill score, while linear QR and QC are worse, as shown in Fig. 3-168.

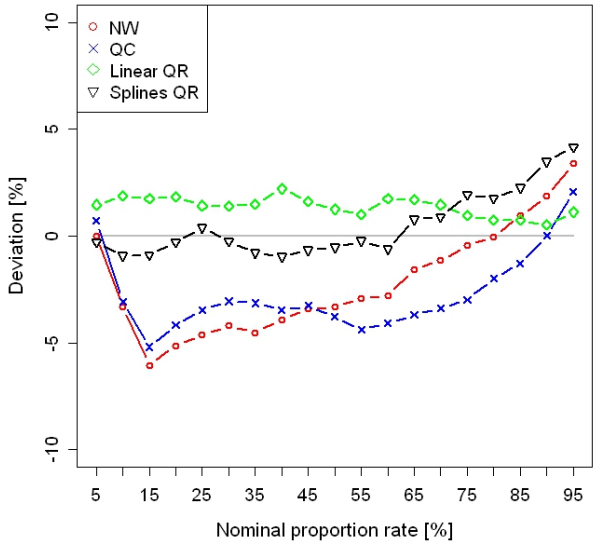

Fig. 3-165 Calibration diagram for the offline test with WFA dataset A.

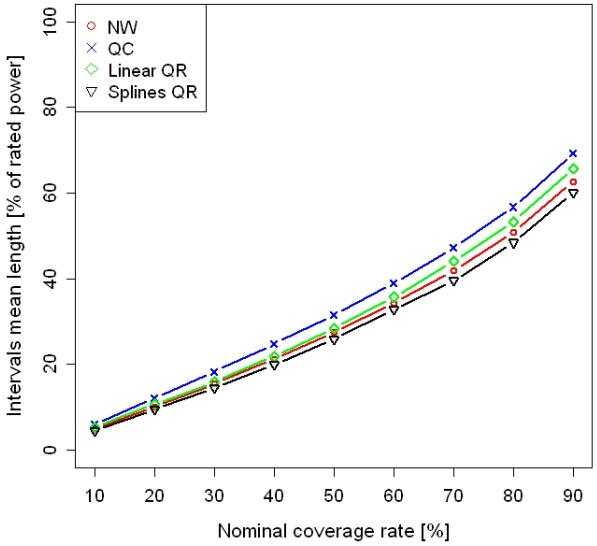

Fig. 3-166 Sharpness diagram for the offline test with WFA dataset A.

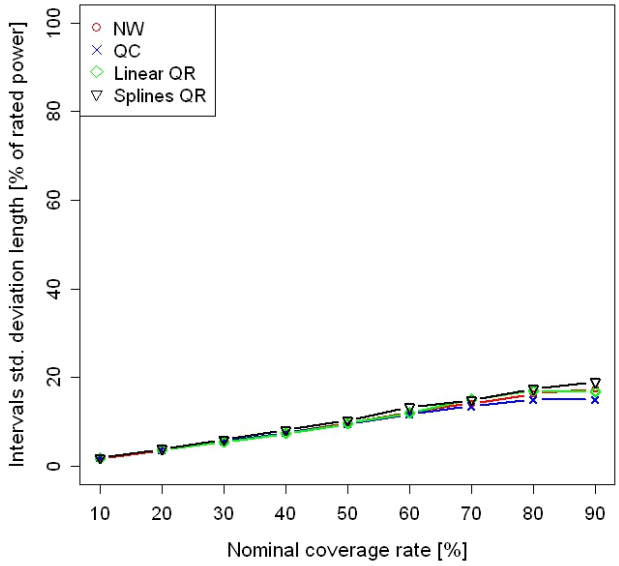

Fig. 3-167 Resolution diagram for the offline test with WFA dataset A.

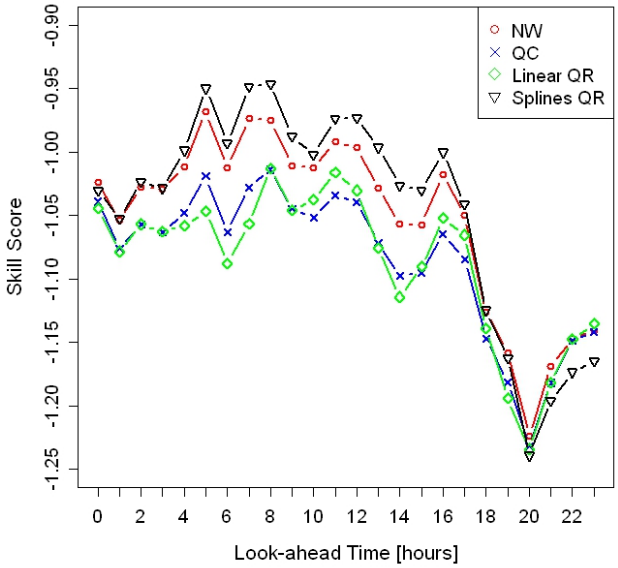

Fig. 3-168 Skill score diagram for the offline test with WFA dataset A.

Fig. 3-169 through Fig. 3-172 depict the results for WFB. The graphs show that the behavior is similar as in the WFA case, except for the calibration, where KDF estimators have better overall calibration than splines QR, with linear QR being the approach with the worst performance and QC the best.

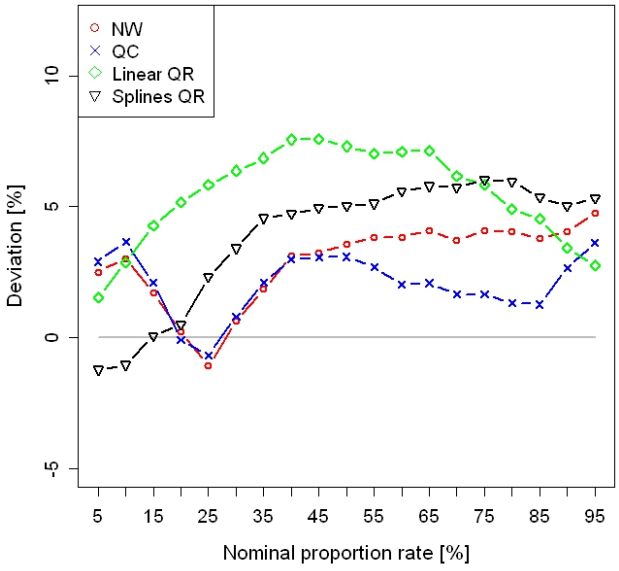

Fig. 3-169 Calibration diagram for the offline test with WFB dataset A.

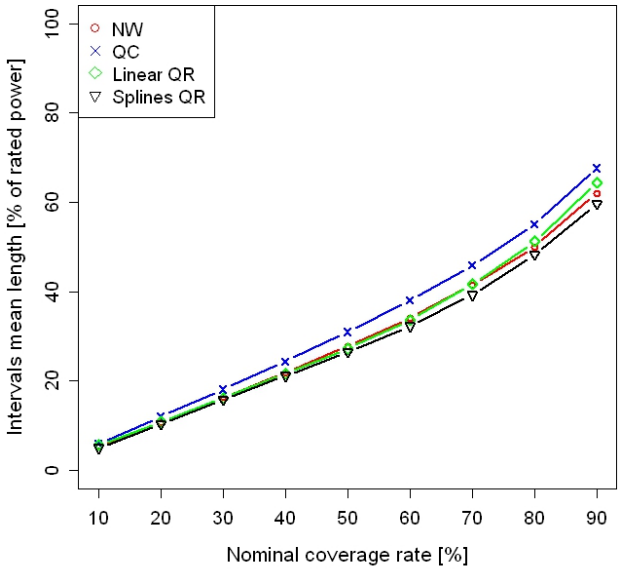

Fig. 3-170 Sharpness diagram for the offline test with WFB dataset A.

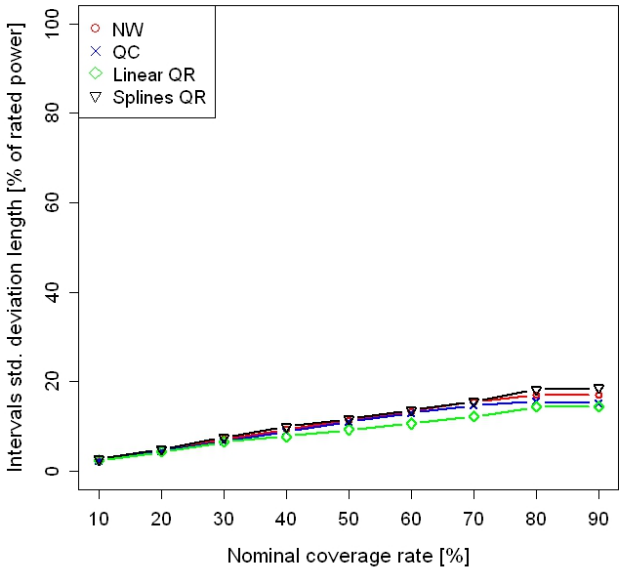

Fig. 3-171 Resolution diagram for the offline test with WFB dataset A.

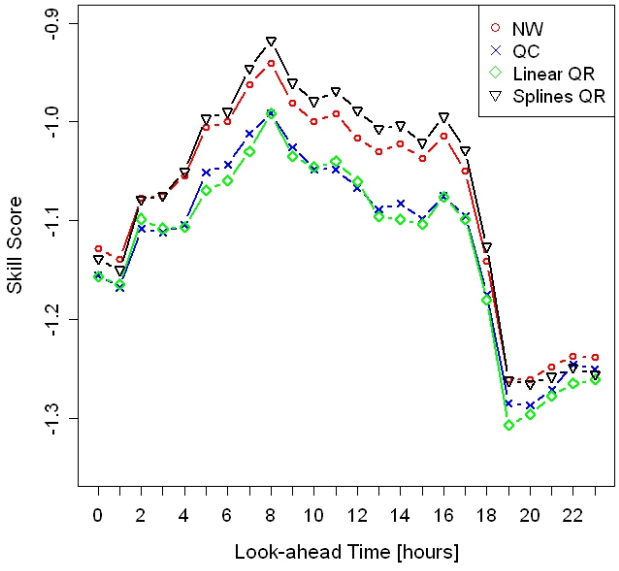

Fig. 3-172 Skill score diagram for the offline test with WFB dataset A.

The main conclusions for this kernel size choice are that splines QR has the best overall calibration in WFA, while KDF approaches perform better in WFB; splines QR has the best

sharpness and resolution; and in terms of the skill score, splines QR has the best performance, followed by NW, and linear QR and QC estimators are the worst.

Kernel size: $(h_{Power}; h_{WindSpeed}) = (0.008; 1)$

Fig. 3-173 depicts the calibration obtained for WFA using various estimators. This graph shows that QR approaches have better calibration results and that KDF estimators tend to underestimate the quantiles.

Fig. 3-174 presents sharpness and Fig. 3-175 resolution. In both, splines QR has the best performance and QC the worst.

The splines QR has better performance in terms of the skill score, while QC has the worst, as shown in Fig. 3-176.

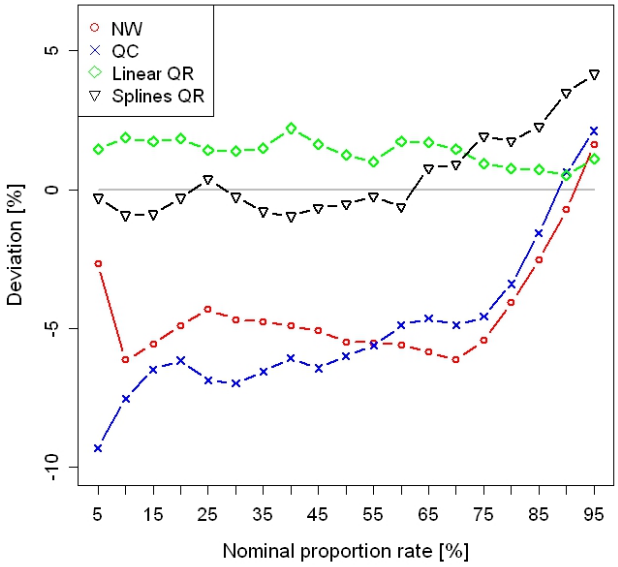

Fig. 3-173 Calibration diagram for the offline test with WFA dataset A.

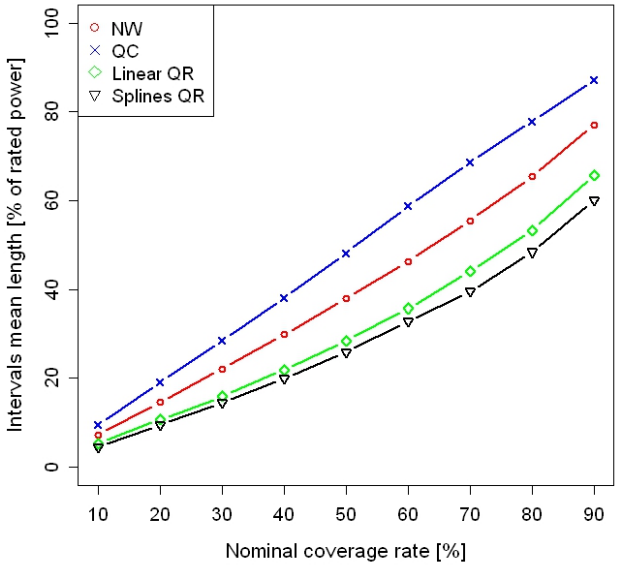

Fig. 3-174 Sharpness diagram for the offline test with WFA dataset A.

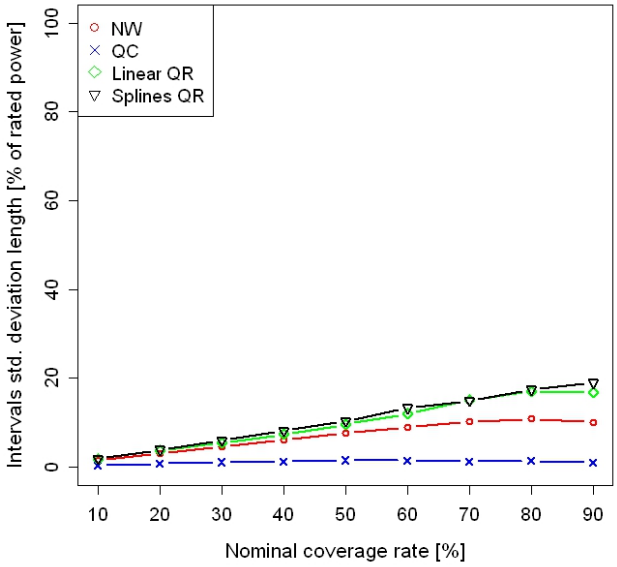

Fig. 3-175 Resolution diagram for the offline test with WFA dataset A.

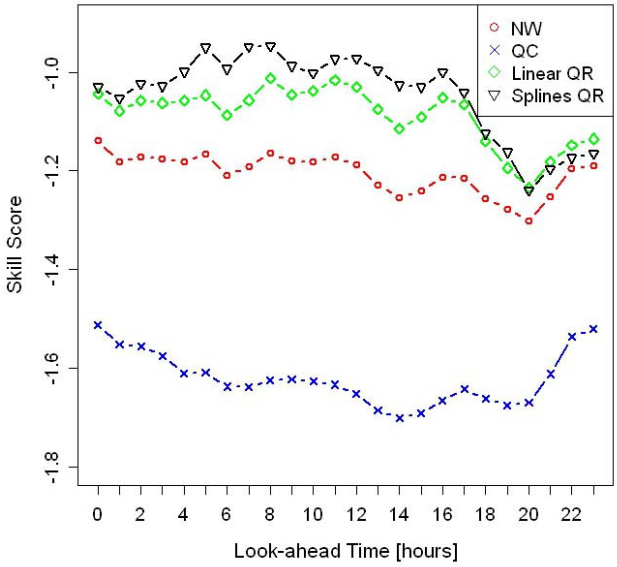

Fig. 3-176 Skill score diagram for the offline test with WFA dataset A.

Fig. 3-177 through Fig. 3-180 depict the results for WFB. The graphs show that the behavior is similar as in the WFA case, except for the calibration, where for quantiles above 25%, KDF estimators become better performers than do splines QR, with linear QR being the approach with the worst calibration performance and NW the best.

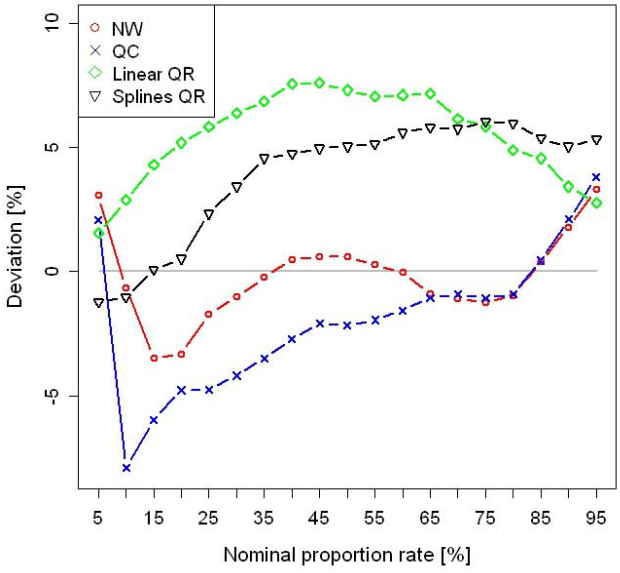

Fig. 3-177 Calibration diagram for the offline test with WFB dataset A.

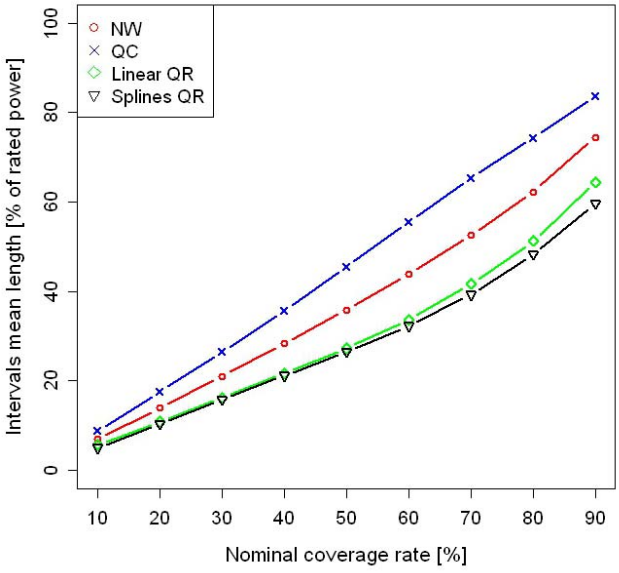

Fig. 3-178 Sharpness diagram for the offline test with WFB dataset A.

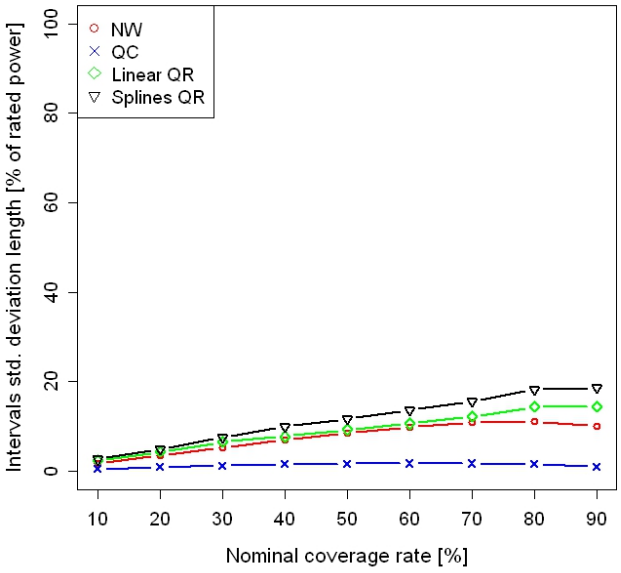

Fig. 3-179 Resolution diagram for the offline test with WFB dataset A.

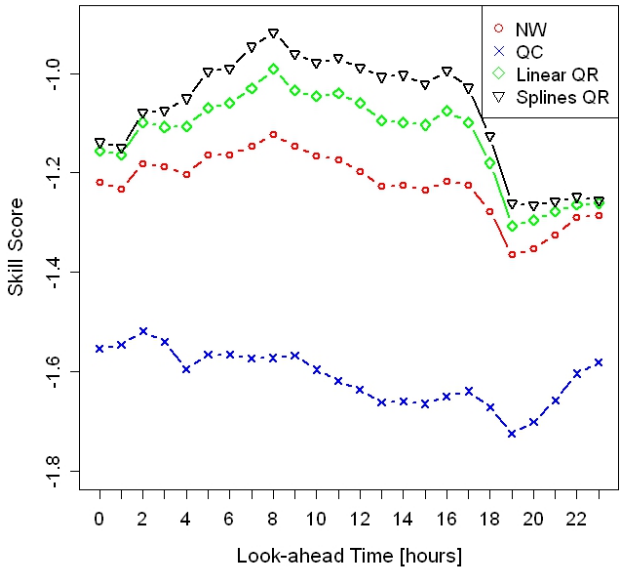

Fig. 3-180 Skill score diagram for the offline test with WFB dataset A.

The main conclusions for this kernel size choice are that splines QR has the best overall calibration in WFA, although in WFB, KDF approaches perform better; splines QR has the best sharpness, and QC the worst resolution; and in terms of the skill score, QR approaches have the best performance and QC is the worst.

Kernel size: $(h_{Power}; h_{WindSpeed}) = (0.01; 1.2)$

Fig. 3-181 depicts the calibration obtained for WFA using various estimators. This graph shows that QR approaches have better calibration results and that KDF estimators tend to underestimate the quantiles.

Fig. 3-182 presents sharpness and Fig. 3-183 resolution. In both, splines QR has the best performance and QC the worst.

The splines QR has better performance in terms of the skill score, while linear QC has the worst, as shown in Fig. 3-184.

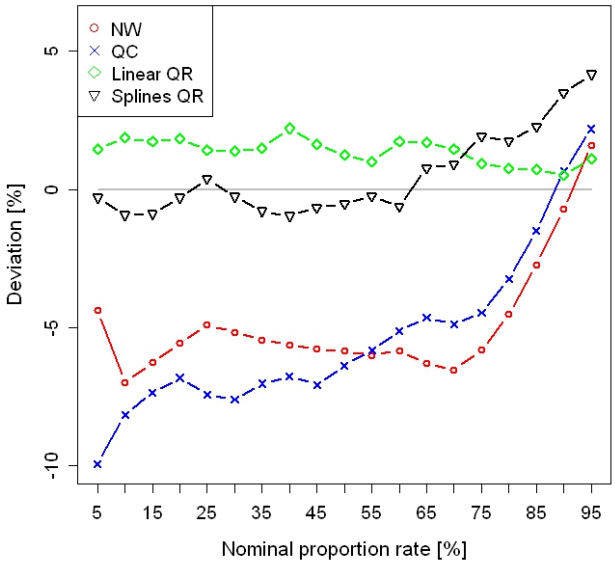

Fig. 3-181 Calibration diagram for the offline test with WFA dataset A.

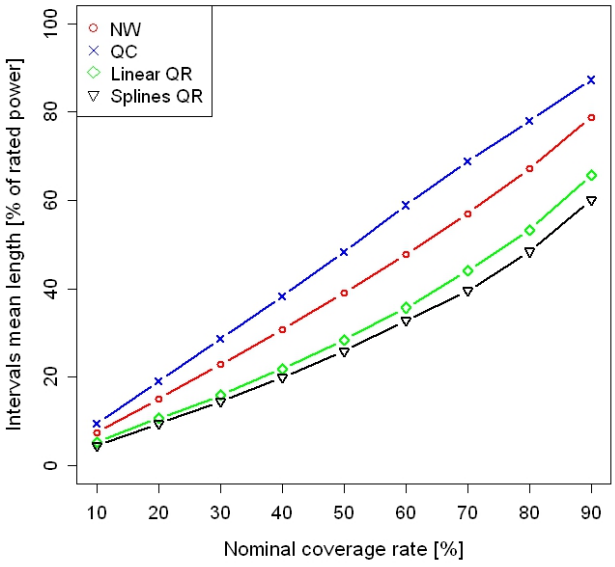

Fig. 3-182 Sharpness diagram for the offline test with WFA dataset A.

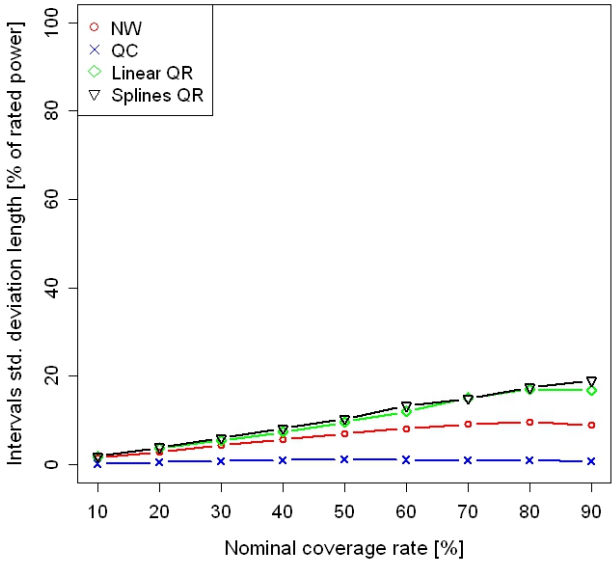

Fig. 3-183 Resolution diagram for the offline test with WFA dataset A.

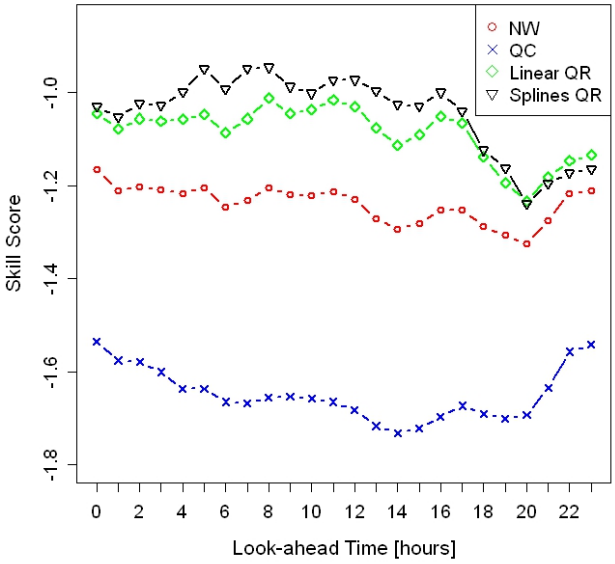

Fig. 3-184 Skill score diagram for the offline test with WFA dataset A.

Fig. 3-185 through Fig. 3-188 depict the results for WFB. The graphs show that the behavior is similar as in the WFA case, except for the calibration, where KDF estimators have better overall performance than do splines QR, with linear QR being the approach with the worst calibration results and NW the best.

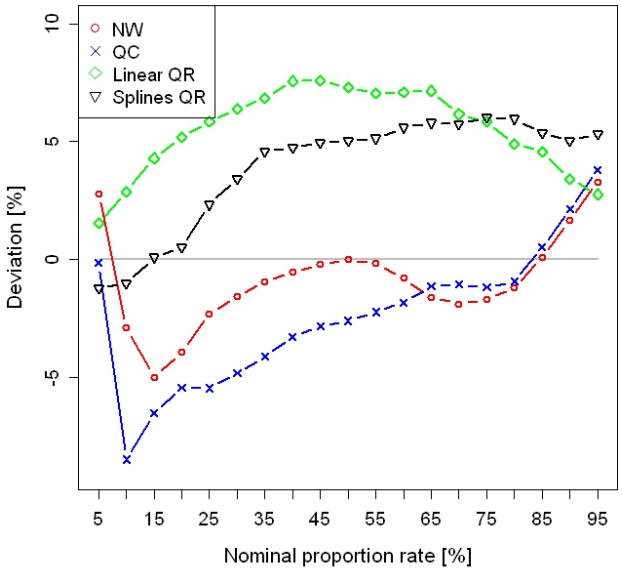

Fig. 3-185 Calibration diagram for the offline test with WFB dataset A.

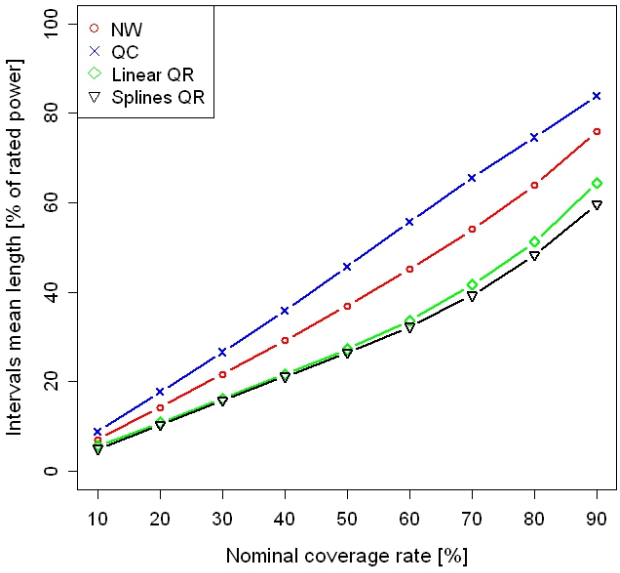

Fig. 3-186 Sharpness diagram for the offline test with WFB dataset A.

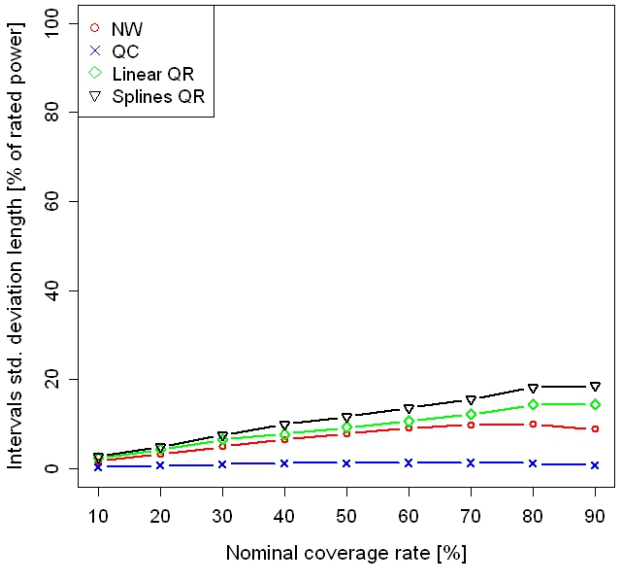

Fig. 3-187 Resolution diagram for the offline test with WFB dataset A.

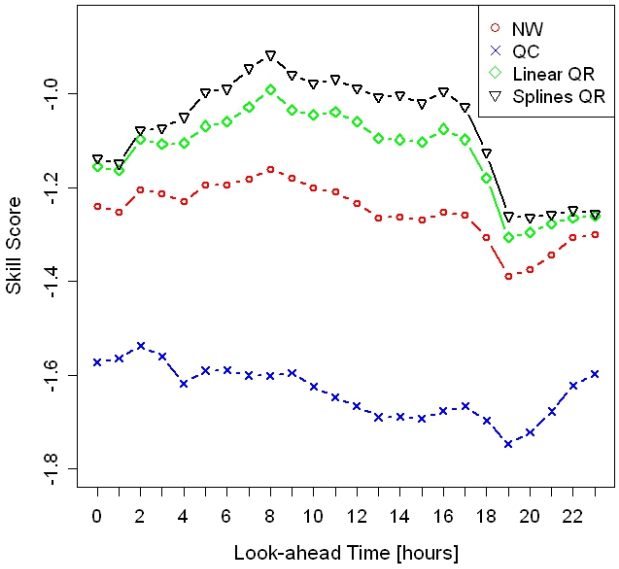

Fig. 3-188 Skill score diagram for the offline test with WFB dataset A.

The main conclusions for this kernel size choice are that splines QR has the best overall calibration in WFA, although KDF approaches perform better in WFB; splines QR has the best

sharpness and resolution; and in terms of the skill score, QR approaches have the best performance and QC is the worst.

Kernel size: $(h_{Power;}\ h_{WindSpeed}) = (0.004;\ 1)$

Fig. 3-189 depicts the calibration obtained for WFA using various estimators. This graph shows that QR approaches have better calibration results and that KDF estimators tend to underestimate the quantiles.

Fig. 3-190 presents sharpness and Fig. 3-191 resolution. In both, splines QR has the best performance and QC the worst.

The splines QR has better performance in terms of the skill score, while QC has the worst, as shown in Fig. 3-192.

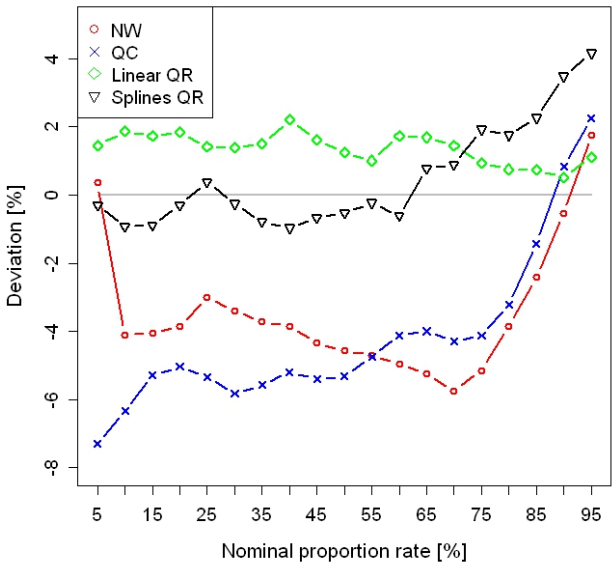

Fig. 3-189 Calibration diagram for the offline test with WFA dataset A.

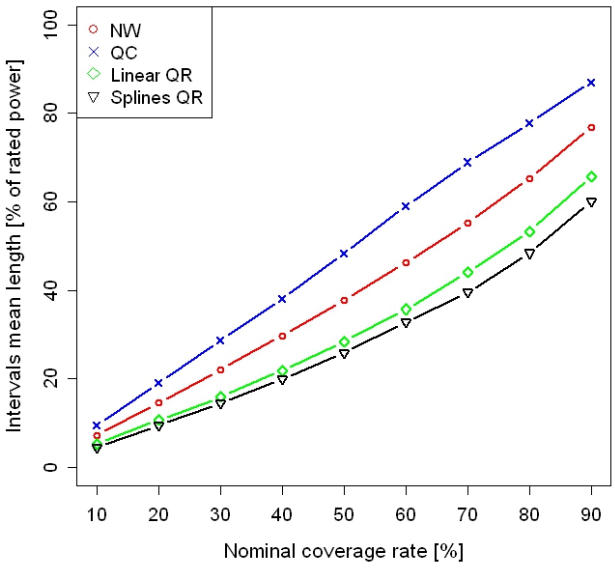

Fig. 3-190 Sharpness diagram for the offline test with WFA dataset A.

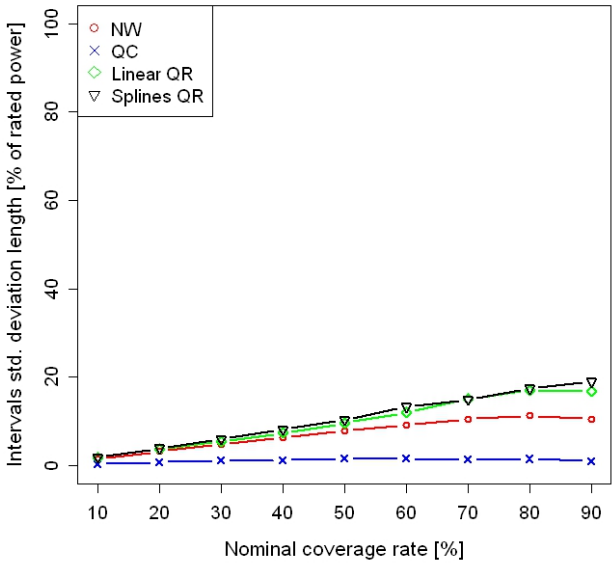

Fig. 3-191 Resolution diagram for the offline test with WFA dataset A.

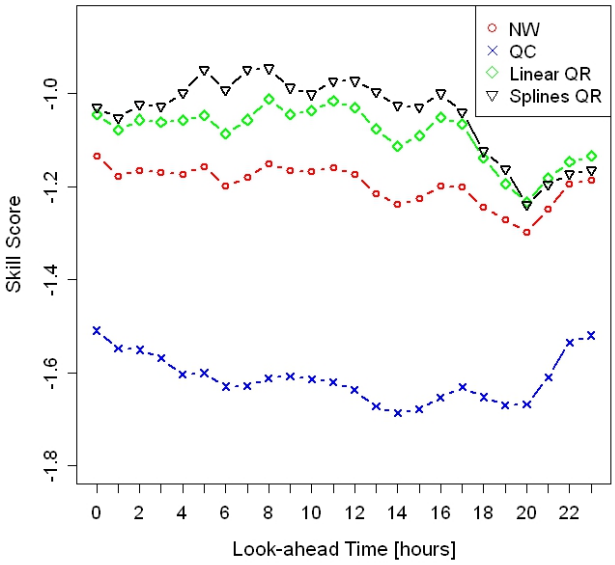

Fig. 3-192 Skill score diagram for the offline test with WFA dataset A.

Fig. 3-193 through Fig. 3-196 depict the results for WFB. The graphs show that the behavior is similar as in the WFA case, except for the calibration, where KDF estimators have better overall calibration than splines QR, with linear QR being the approach with the worst performance and NW the best.

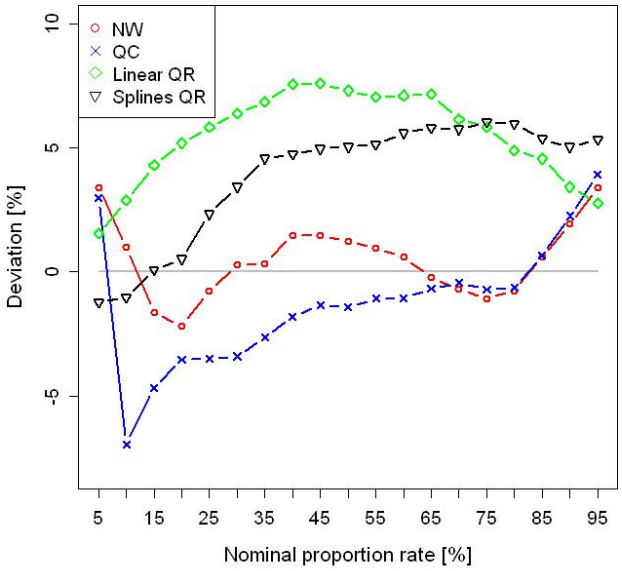

Fig. 3-193 Calibration diagram for the offline test with WFB dataset A.

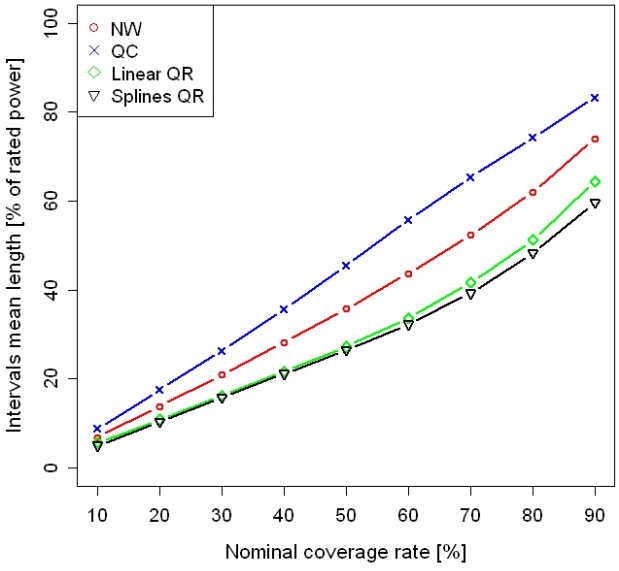

Fig. 3-194 Sharpness diagram for the offline test with WFB dataset A.

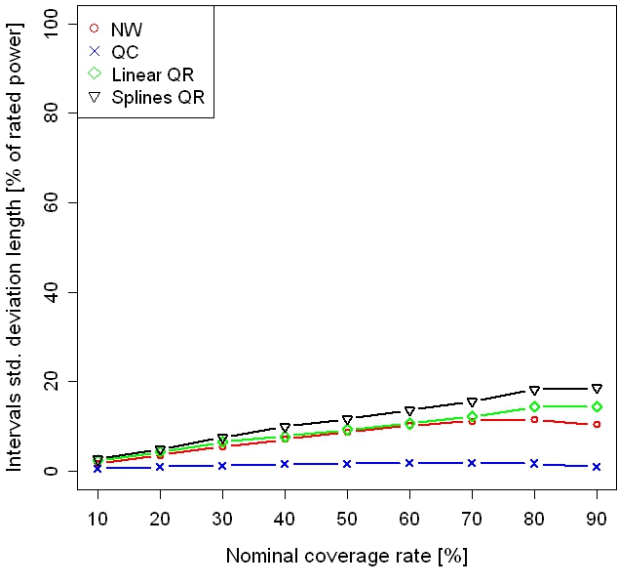

Fig. 3-195 Resolution diagram for the offline test with WFB dataset A.

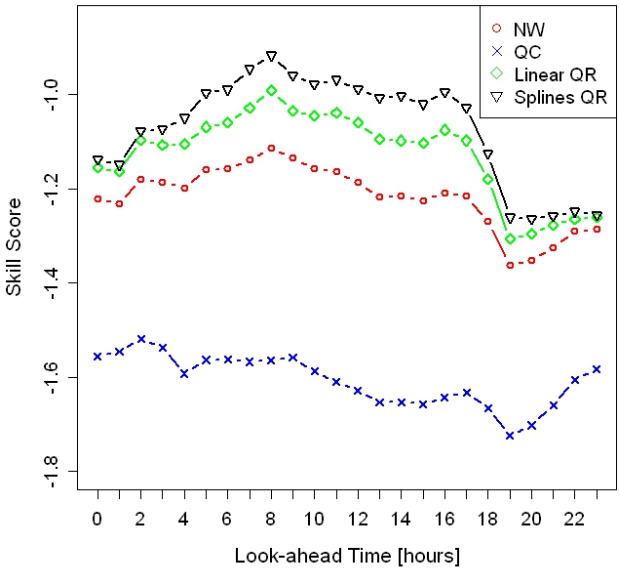

Fig. 3-196 Skill score diagram for the offline test with WFB dataset A.

The main conclusions for this kernel size choice are that splines QR has the best overall calibration in WFA, while KDF approaches perform better in WFB; splines QR has the best sharpness and resolution; and in terms of the skill score, QR approaches have the best performance and the QC estimators are the worst.

Kernel Sizes Results

For wind speed kernel sizes smaller than 1, QR estimators perform better in WFA, while in WFB, KDF approaches have the best calibration, particularly for quantiles above 30%. Sharpness, resolution, and skill score results were similar among the four approaches in both wind farms, such that splines QR has the best sharpness and resolution performance, although the latter is almost the same for all estimators; in terms of the skill score, splines QR is the best, followed by NW, QC, and linear QR competing for the worst performance.

On the other hand, for wind speed kernel sizes equal to or larger than 1, results are worse than they are in the previous case, although calibration and sharpness display similar behavior. For both wind farms, QC has the worst resolution and QR approaches have the best skill score performance.

3.4.3.5 48 Hours-Ahead Offline Evaluation Results

In this subsection, the impact of the different variables forecasted by the NWP model will be studied. Moreover, the results with forecasts launched at 6:00 AM and 6:00 PM are also evaluated.

The training dataset for both wind farms was selected to have 70% of all examples (30% of examples for testing). The following training and testing datasets were considered:

- **Wind farm A (WFA):** Training set ran from January 1, 2009, to November 21, 2009 (12,169 points), and the testing set ran from November 22, 2009, to February 20, 2010 (5,203 points);
- **Wind Farm B (WFB):** Training set ran from January 1, 2009, to November 18, 2009 (12,384 points), and the testing set ran from November 19, 2009, to February 20, 2010 (5,332 points).

The following models were considered and compared: (M0) wind speed; (M1) wind speed + direction; (M2) wind speed + hour of the day; (M3) wind speed + look-ahead time step; (M4) wind speed + direction + hour of the day; and (M5) wind speed + direction + look-ahead time step.

Wind Farm A

Nadaraya-Watson (NW) KDF

The following kernel functions were used in the NW estimator and WFA:

- Wind power generation: Chen's beta kernel from (3-17) with a bandwidth equal to 0.008;
- Wind speed forecast: Chen's gamma kernel from (3-21) with a bandwidth equal to 0.05;
- Wind direction: von Mises distribution from (3-24) with a bandwidth equal to 2.5;
- Look-ahead time step: Chen's beta kernel from (3-17) with a bandwidth equal to 0.1;
- Hour of the day: von Mises distribution from (3-24) with a bandwidth equal to 2.5.

The kernel bandwidth values were determined experimentally (via trial and error) and using as a starting point the values suggested by the function *cde.bandwidths* from the R package "hdrcde" [58].

Fig. 3-197 depicts the calibration obtained with an offline approach for WFA and using NWP launched at 6 AM, while Fig. 3-198 depicts the calibration obtained with NWP launched at 6 PM. The calibration between the two figures is slightly different: the NWP from 6 AM presents a better performance, in particular for quantiles below 50%. From the five different models (M0–M5), the best performance is from models M3 (wind speed + look-ahead time step) and M0 (wind speed) for both studies. Nevertheless, the performance of all methods is rather similar, and there are no significant differences.

Note that the inclusion of direction in the model (M1) decreases the performance in terms of calibration. Because the look-ahead time step seems to improve the calibration performance, in Fig. 3-199 and Fig. 3-200 the calibration diagram is depicted for look-ahead time step *t+6h* obtained with 6 AM and 6 PM NWPs. The calibration for *t+15h* can be found in Appendix B. The results show, on overall terms, a best performance for models M2 and M3.

An interesting result can be found in Appendix B for look-ahead time step *t+15h*. The model M2 and M4 are overestimating the quantiles, while the other methods are underestimating.

Fig. 3-201 and Fig. 3-202 depict the sharpness obtained for WFA with 6 AM and 6 PM NWP. The sharpness is rather similar for all models; the same is true when it is computed for a specific look-ahead time step, such as *t+6h* in Fig. 3-203 and Fig. 3-204, or *t+15h* in Appendix B. Nevertheless, models M4 and M5 present the best sharpness performance, while M1 presents the lowest performance.

Fig. 3-205 and Fig. 3-206 depict the resolution obtained for WFA. In this case, the model M4 presents the best performance for both 6 AM and 6 PM NWPs. The worst performance is from M0. When the analysis is performed for the look-ahead time step *t+6h* (Fig. 3-207 and Fig. 3-208), the model with best performance is M1; however, for *t+15h* (see Appendix B), the best performance is from M4. Hence, on overall terms, the best resolution performance is from model M4.

Fig. 3-209 and Fig. 3-210 present the skill score computed for each look-ahead time step with (3-46) for 6 AM and 6 PM NWPs. The performance of all of these models is rather similar, with a slight advantage for models M2 and M3 at both 6 AM and 6 PM.

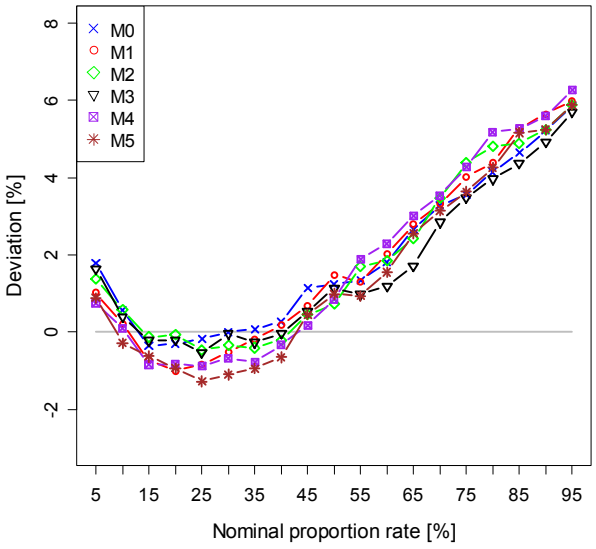

Fig. 3-197 Calibration diagram for WFA with 6:00 AM NWP and NW models M0–M5.

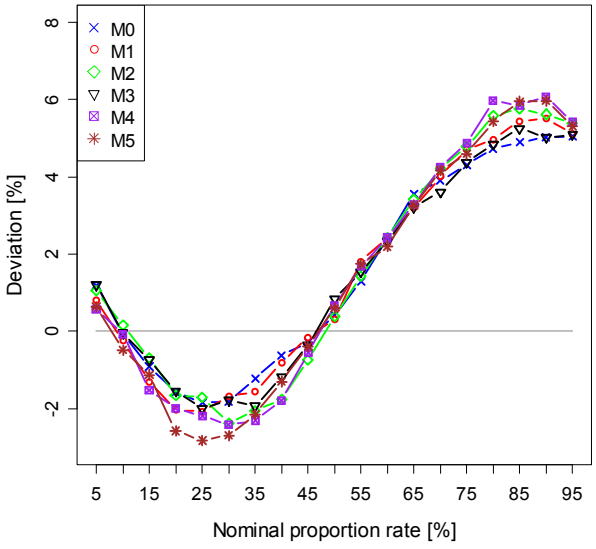

Fig. 3-198 Calibration diagram for WFA with 6:00 PM NWP and NW models M0–M5.

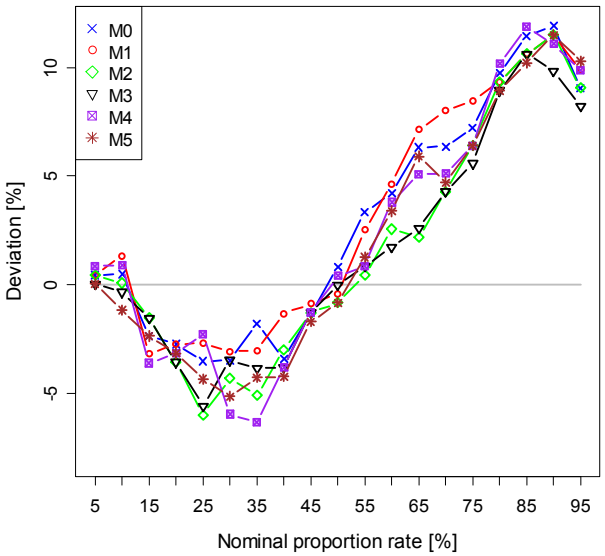

Fig. 3-199 Calibration diagram for WFA with 6:00 AM NWP and NW models M0–M5 for look-ahead time step t+6h.

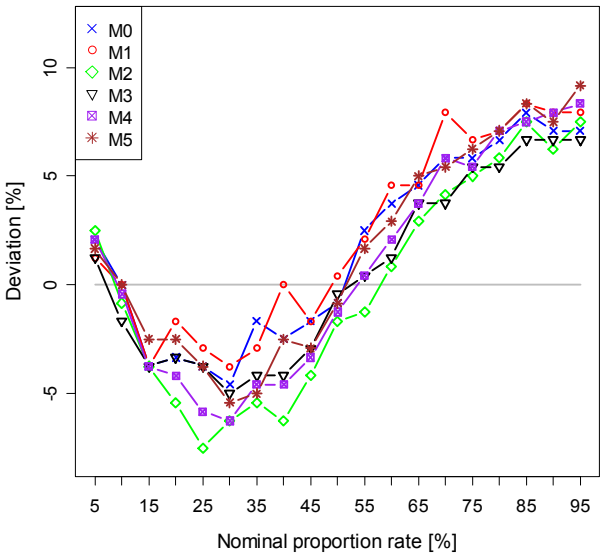

Fig. 3-200 Calibration diagram for WFA with 6:00 PM NWP and NW models M0–M5 for look-ahead time step t+6h.

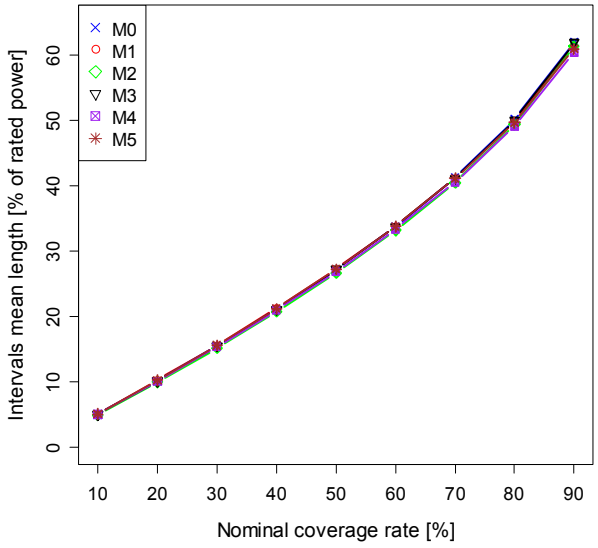

Fig. 3-201 Sharpness diagram for WFA with 6:00 AM NWP and NW models M0–M5.

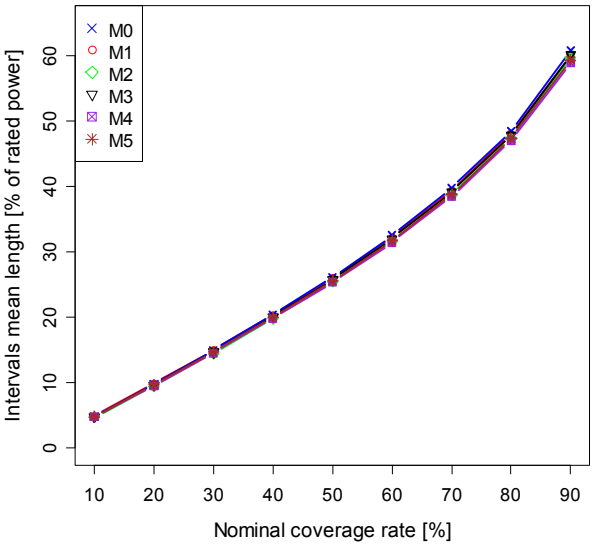

Fig. 3-202 Sharpness diagram for WFA with 6:00 PM NWP and NW models M0–M5.

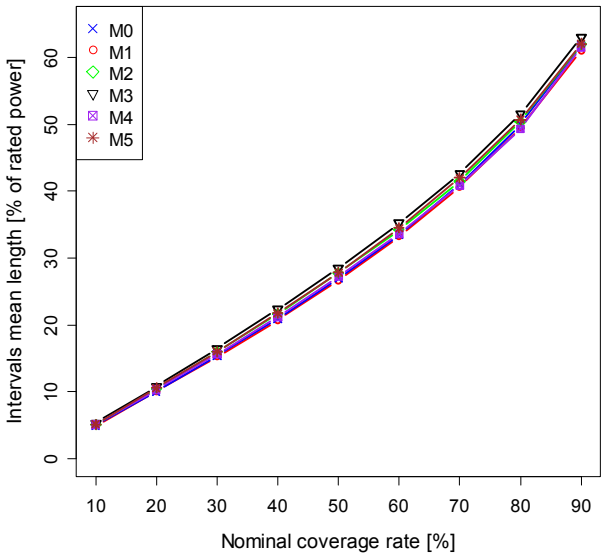

Fig. 3-203 Sharpness diagram for WFA with 6:00 AM NWP and NW models M0–M5 for look-ahead time step t+6h.

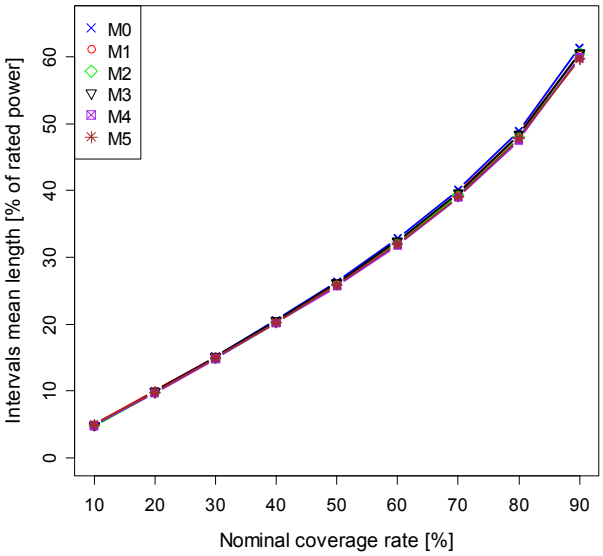

Fig. 3-204 Sharpness diagram for WFA with 6:00 PM NWP and NW models M0–M5 for look-ahead time step t+6h.

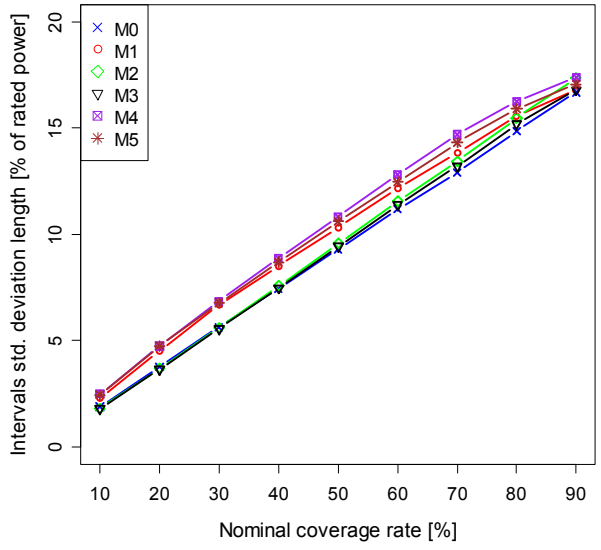

Fig. 3-205 Resolution diagram for WFA with 6:00 AM NWP and NW models M0–M5.

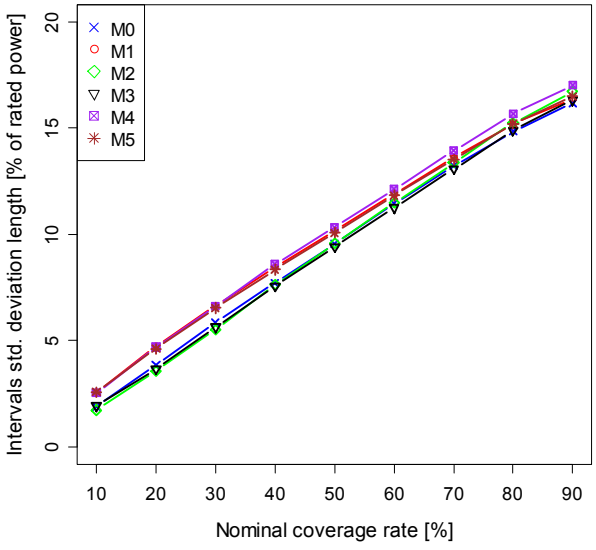

Fig. 3-206 Resolution diagram for WFA with 6:00 PM NWP and NW models M0–M5.

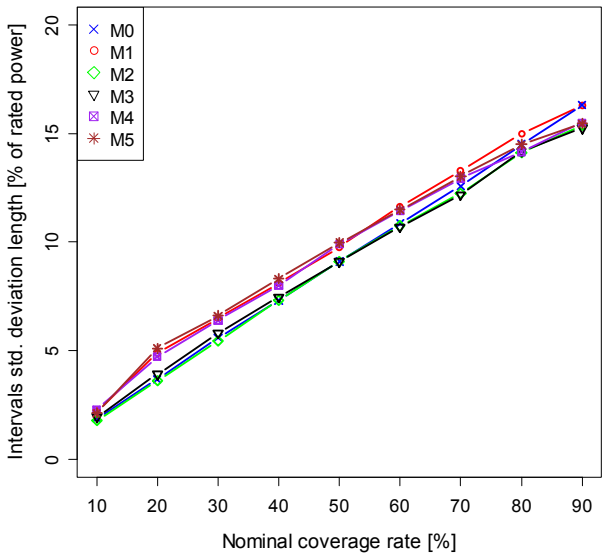

Fig. 3-207 Resolution diagram for WFA with 6:00 AM NWP and NW models M0–M5 for look-ahead time step t+6h.

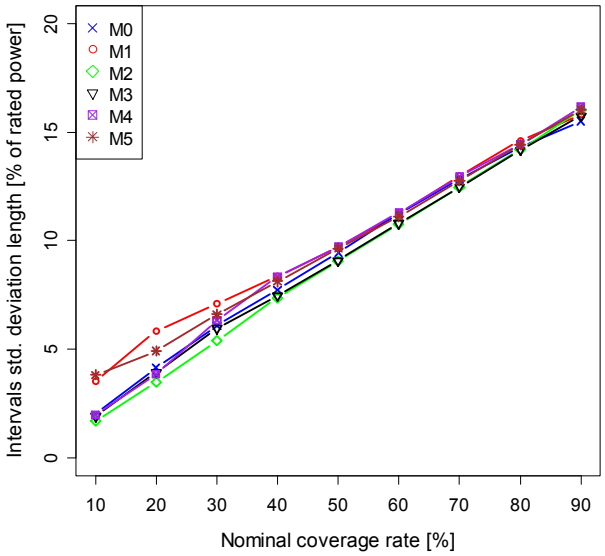

Fig. 3-208 Resolution diagram for WFA with 6:00 PM NWP and NW models M0–M5 for look-ahead time step t+6h.

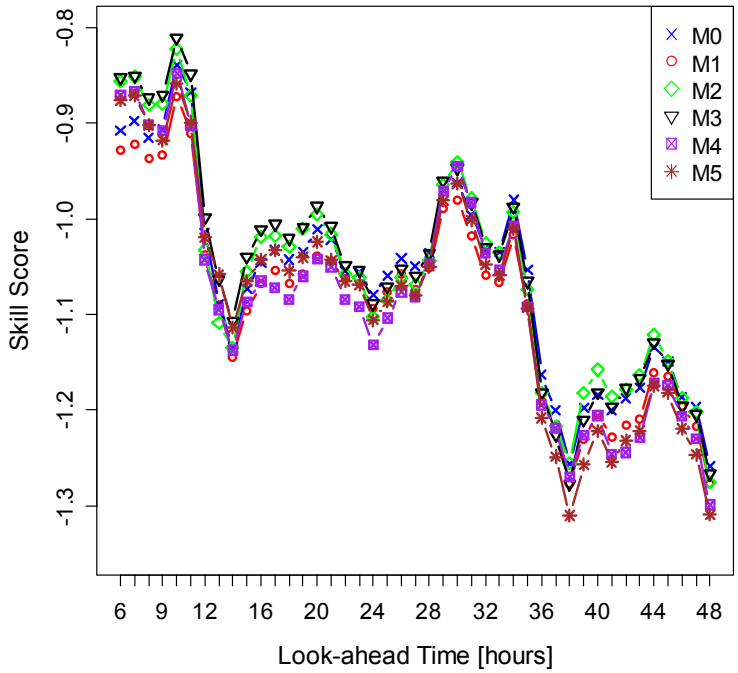

Fig. 3-209 Skill score diagram for WFA with 6:00 AM NWP and NW models M0–M5.

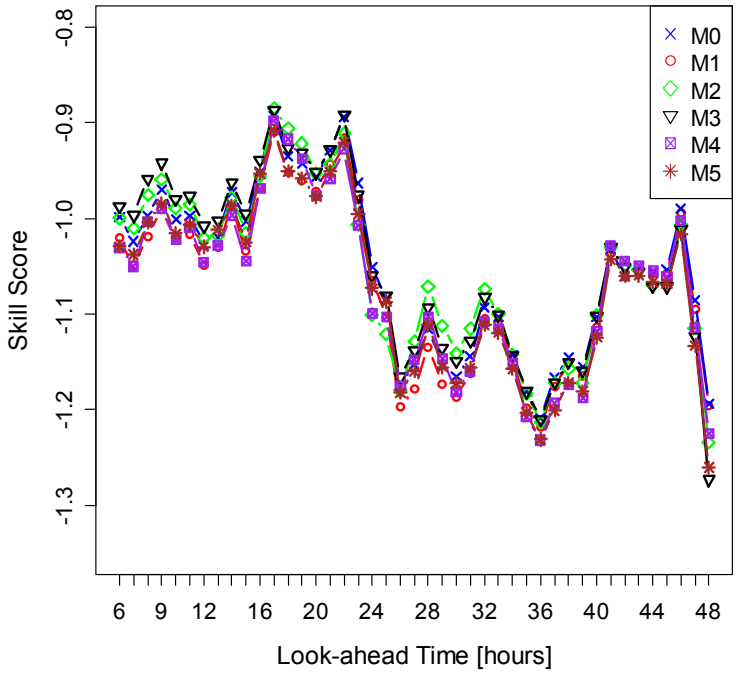

Fig. 3-210 Skill score diagram for WFA with 6:00 PM NWP and NW models M0–M5.

Quantile-Copula (QC) KDF

The following kernel functions were used in the Quantile-Copula (QC) estimator and WFA:

- Wind power generation: Chen's beta kernel from 3-19 with a bandwidth equal to 0.008;
- Wind speed forecast: Chen's beta kernel from 3-19 with a bandwidth equal to 0.008;
- Wind direction: von Mises distribution from 3-26 with a bandwidth equal to 1.0;
- Look-ahead time step: Chen's beta kernel from 3-19 with a bandwidth equal to 0.2;
- Hour of the day: von Mises distribution from 3-26 with a bandwidth equal to 1.0.

The kernel bandwidth values were determined experimentally (via trial and error) and using as a starting point the values suggested by the function *cde.bandwidths* from the R package "hdrcde" [58].

Fig. 3-211 and Fig. 3-212 depict the calibration obtained with an offline approach for WFA and using NWP launched at 6 AM and 6PM, respectively. The calibration between the two figures is slightly different: the NWP from 6 AM presents a better performance, in particular for quantiles below 50%. From the five different models (M0–M5), the best performance is from models M3 (wind speed + look-ahead time step) and M2 (wind speed + hour of the day) for both studies. The performance of all methods is rather similar, and there are no significant differences.

In the QC estimator, the inclusion of wind direction in the model (M1) increases the calibration performance for some quantiles.

Fig. 3-213 and Fig. 3-214 depict the calibration diagram for look-ahead time step *t+6h* obtained with 6 AM and 6 PM NWPs. The calibration for *t+15h* can be found in Appendix B. The results show, on overall terms, a best performance for models M2 and M3.

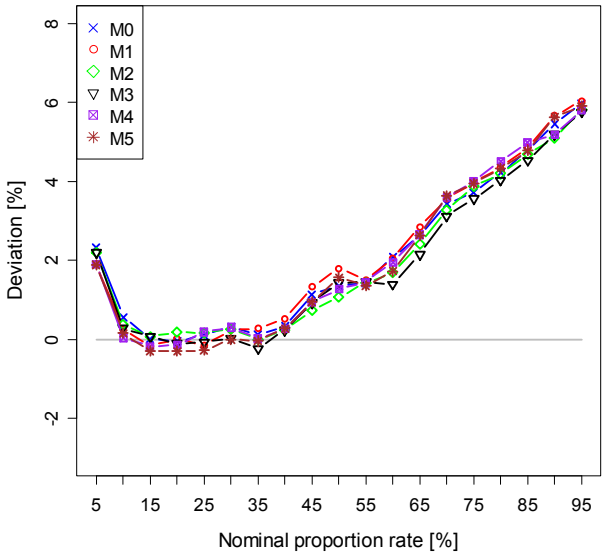

Fig. 3-211 Calibration diagram for WFA with 6:00 AM NWP and QC models M0–M5.

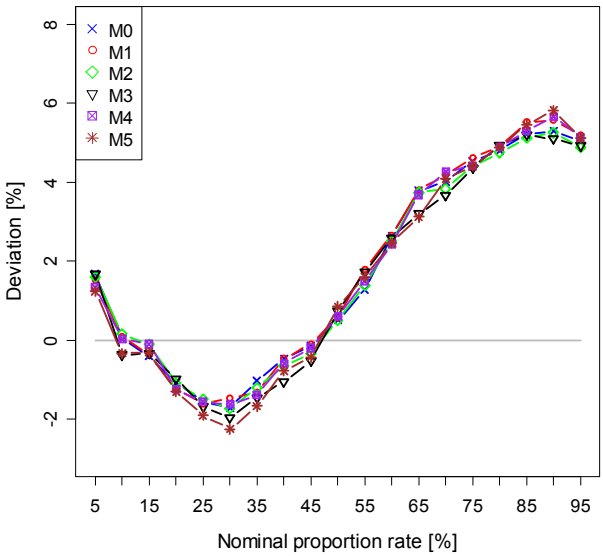

Fig. 3-212 Calibration diagram for WFA with 6:00 PM NWP and QC models M0–M5.

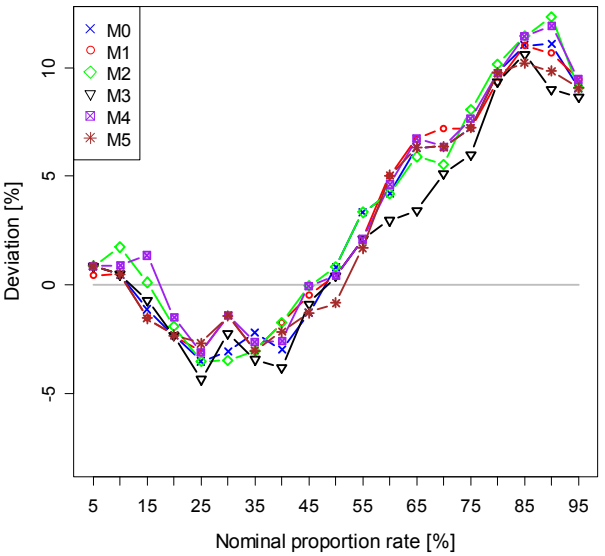

Fig. 3-213 Calibration diagram for WFA with 6:00 AM NWP and QC models M0–M5 for look-ahead time step t+6h.

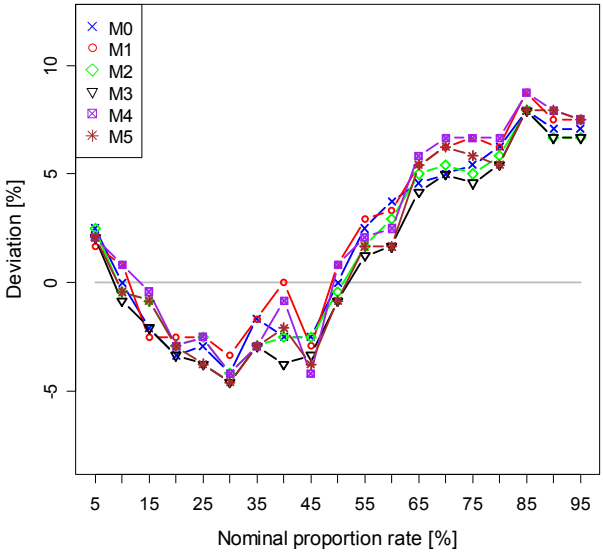

Fig. 3-214 Calibration diagram for WFA with 6:00 PM NWP and QC models M0–M5 for look-ahead time step t+6h.

Fig. 3-215 and Fig. 3-216 depict the sharpness obtained for WFA with 6 AM and 6 PM NWPs. The sharpness is almost equal for all models, even when it is computed for each look-ahead time step (see Fig. 3-217, Fig. 3-218, and Appendix B).

Fig. 3-219 and Fig. 3-220 depict the resolution obtained for WFA. In this case, the model M4 presents the best performance for both 6 AM and 6 PM NWPs, although the performances of M5 and M1 are also very close. The results are verified for look-ahead time step *t+6h* (Fig. 3-221 and Fig. 3-222).

Fig. 3-223 and Fig. 3-224 present the skill score computed for each look-ahead time step and for 6 AM and 6 PM NWPs. The performance of all of these models is very similar, with a slight advantage for models M2 and M3 at both 6 AM and 6 PM.

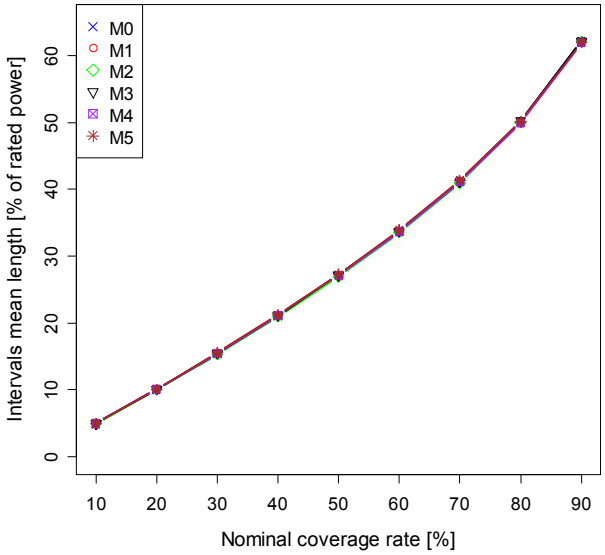

Fig. 3-215 Sharpness diagram for WFA with 6:00 AM NWP and QC models M0–M5.

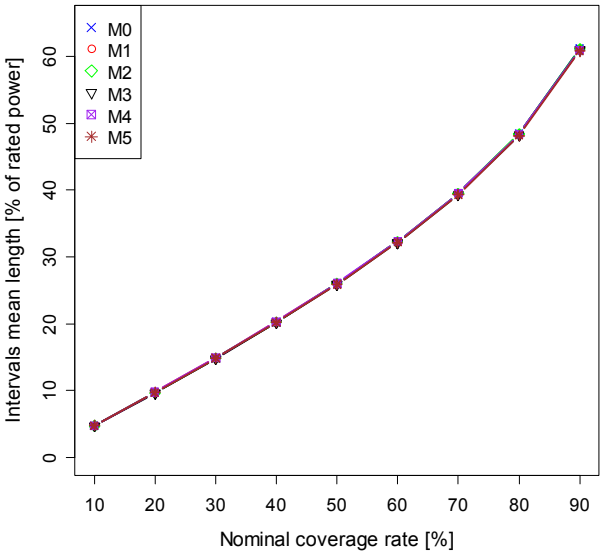

Fig. 3-216 Sharpness diagram for WFA with 6:00 PM NWP and QC models M0–M5.

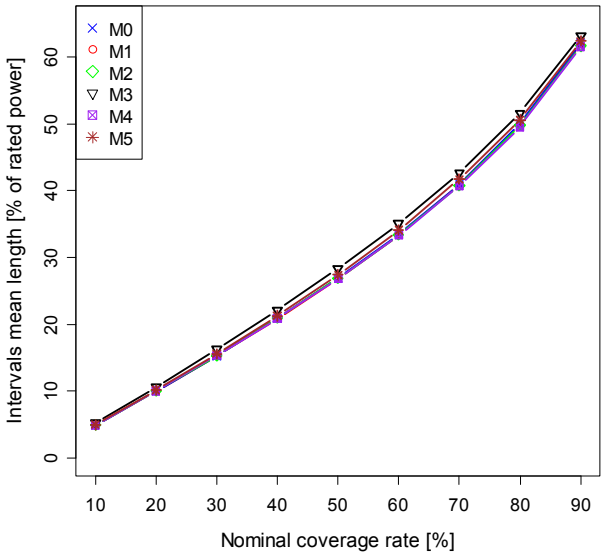

Fig. 3-217 Sharpness diagram for WFA with 6:00 AM NWP and QC models M0–M5 for look-ahead time step t+6h.

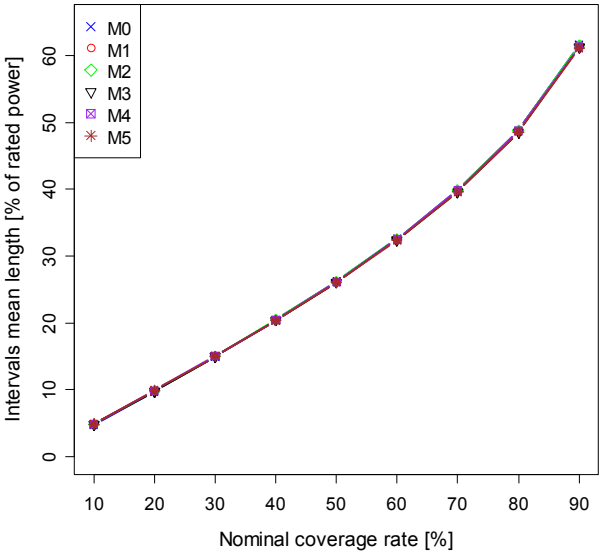

Fig. 3-218 Sharpness diagram for WFA with 6:00 PM NWP and QC models M0–M5 for look-ahead time step t+6h.

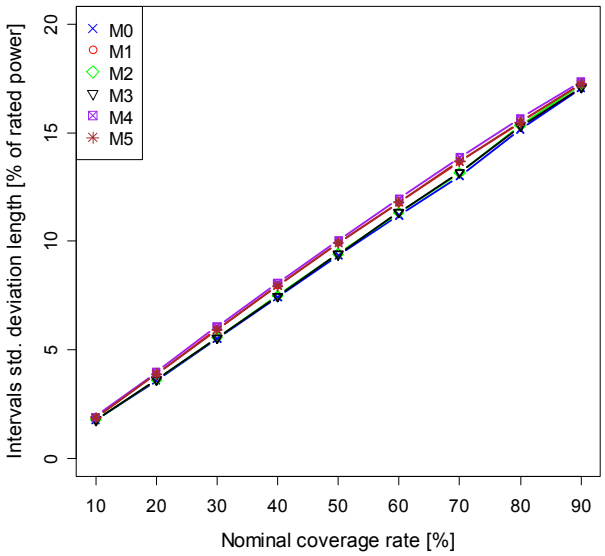

Fig. 3-219 Resolution diagram for WFA with 6:00 AM NWP and QC models M0–M5.

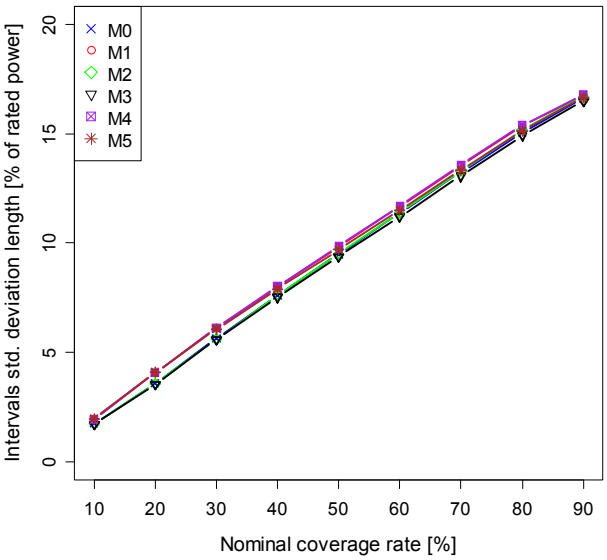

Fig. 3-220 Resolution diagram for WFA with 6:00 PM NWP and QC models M0–M5.

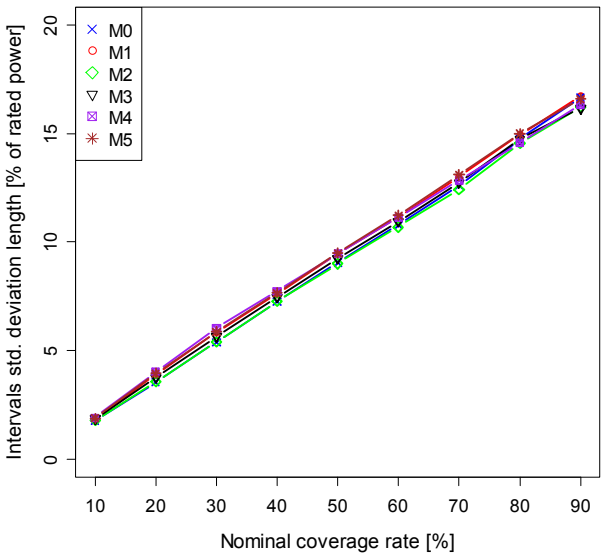

Fig. 3-221 Resolution diagram for WFA with 6:00 AM NWP and QC models M0–M5 for look-ahead time step t+6h.

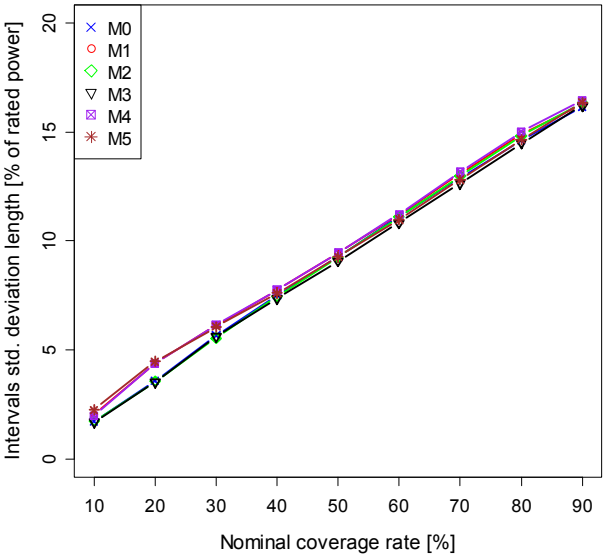

Fig. 3-222 Resolution diagram for WFA with 6:00 PM NWP and QC models M0–M5 for look-ahead time step t+6h.

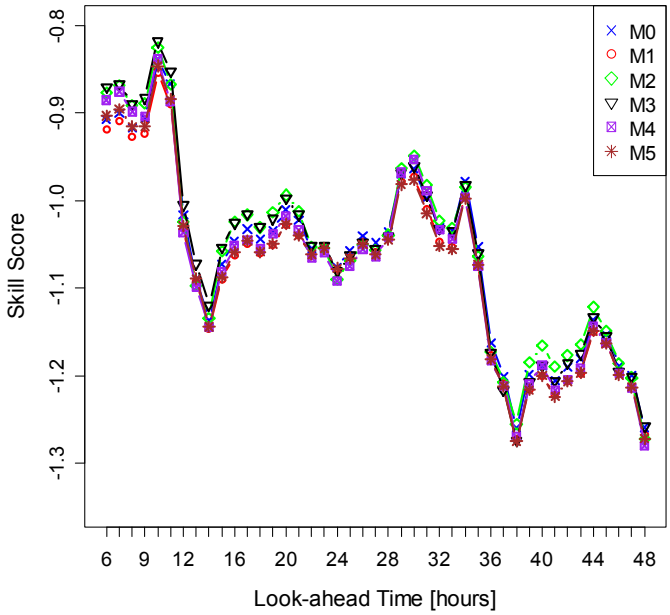

Fig. 3-223 Skill score diagram for WFA with 6:00 AM NWP and QC models M0–M5.

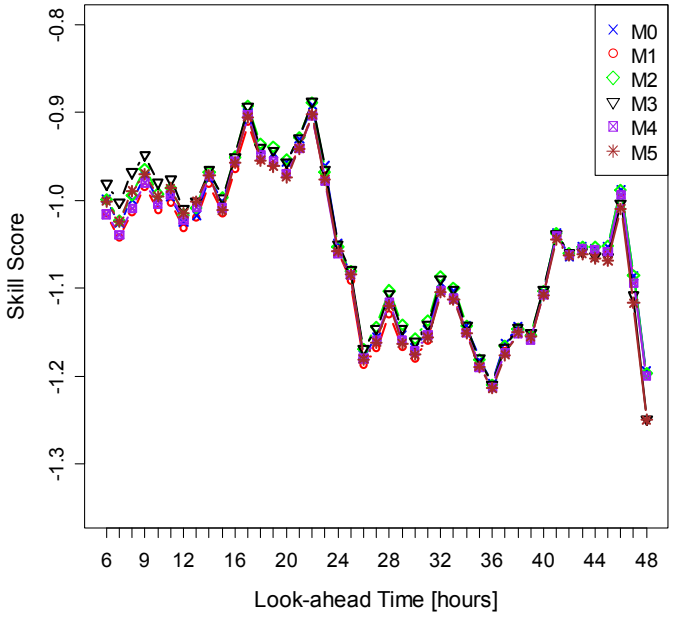

Fig. 3-224 Skill score diagram for WFA with 6:00 PM NWP and QC models M0–M5.

Splines Quantile Regression (splines QR)

For the circular variables, such as direction and hour of the day, a periodic cubic spline basis with equidistant knots is used. This is carried out by the S-PLUS/R functions pb.bse, pb.h, and bint0 available at http://www.imm.dtu.dk/~han/pub.

Fig. 3-225 and Fig. 3-226 depict the calibration obtained with an offline approach for WFA and using NWP launched at 6 AM and 6 PM, respectively. As with the other methods, the calibration

between the two figures is slightly different: the NWP from 6 AM presents a better performance, in particular for quantiles below 50%. For this method, the models with best calibration performance are M0 and M1; however, the overall performance is generally very similar.

Also for this estimator, the inclusion of wind direction increases the calibration performance for some quantiles.

Fig. 3-227 and Fig. 3-228 depict the calibration diagram for look-ahead time step *t+6h* obtained with 6 AM and 6 PM NWPs. For this look-ahead time step, it is very difficult to find the model with the best performance.

Fig. 3-229 and Fig. 3-230 depict the sharpness obtained for WFA with 6 AM and 6 PM NWPs. The sharpness is almost equal for all models, even when it is computed for each look-ahead time step (Fig. 3-231 and Fig. 3-232).

Fig. 3-233 and Fig. 3-234 depict the resolution obtained for WFA. In this case, the models from M2–M5 present the best performance for both 6 AM and 6 PM NWPs. For look-ahead time step *t+6h*, it is the reverse situation, as depicted in Fig. 3-235 and Fig. 3-236.

Fig. 3-237 and Fig. 3-238 present the skill score computed for each look-ahead time step and for 6 AM and 6 PM NWPs. The performance of all of these models is very similar, with a slight advantage for models M2–M5 at both 6 AM and 6 PM.

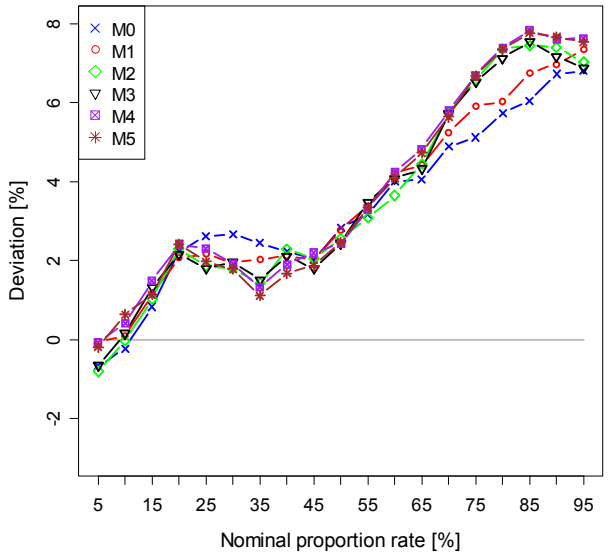

Fig. 3-225 Calibration diagram for WFA with 6:00 AM NWP and splines QR models M0–M5.

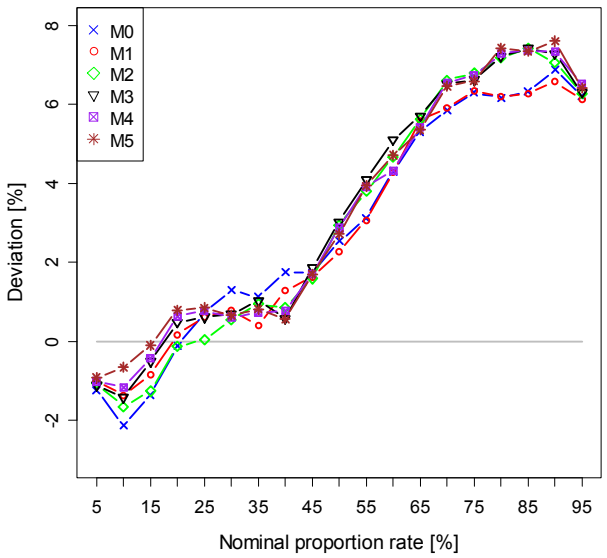

Fig. 3-226 Calibration diagram for WFA with 6:00 PM NWP and splines QR models M0–M5.

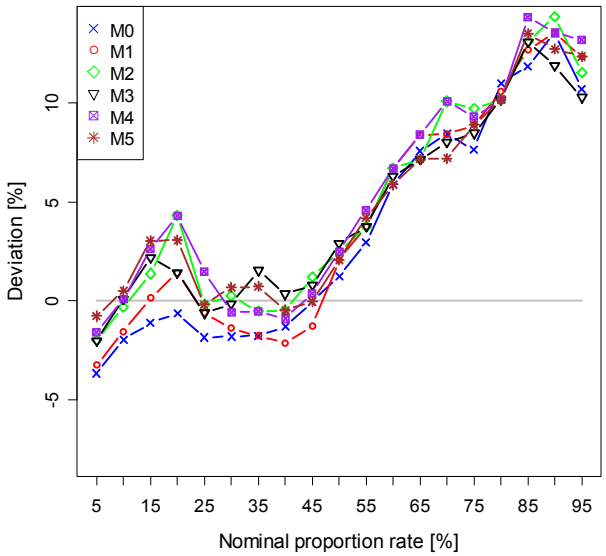

Fig. 3-227 Calibration diagram for WFA with 6:00 AM NWP and splines QR models M0–M5 for look-ahead time step t+6h.

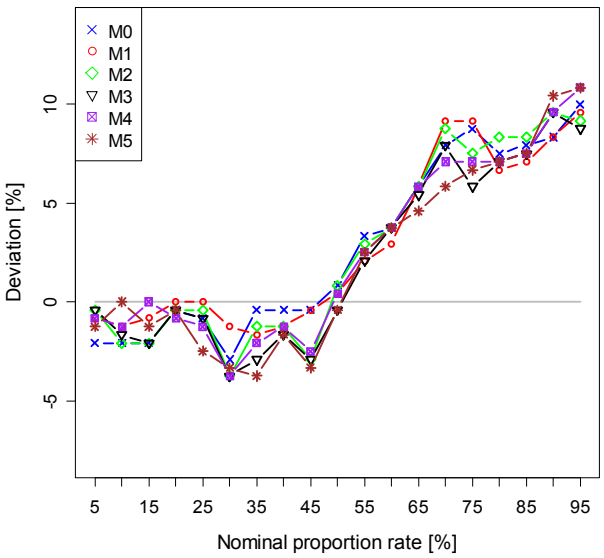

Fig. 3-228 Calibration diagram for WFA with 6:00 PM NWP and splines QR models M0–M5 for look-ahead time step t+6h.

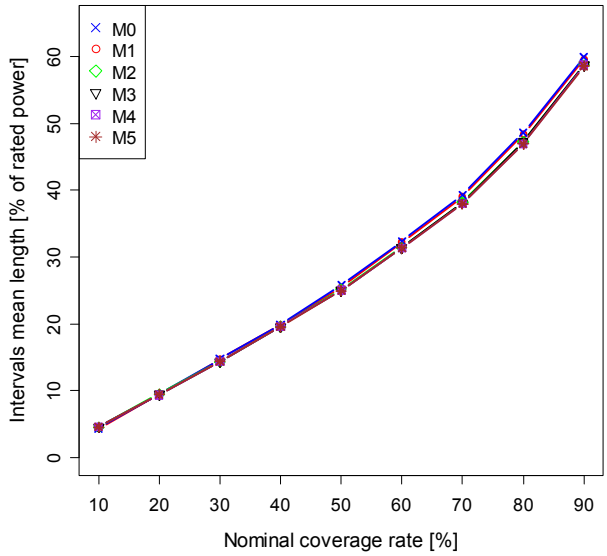

Fig. 3-229 Sharpness diagram for WFA with 6:00 AM NWP and splines QR models M0–M5.

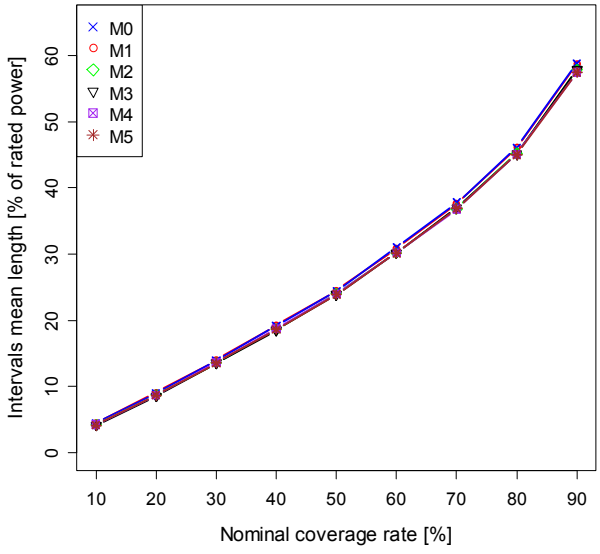

Fig. 3-230 Sharpness diagram for WFA with 6:00 PM NWP and splines QR models M0–M5.

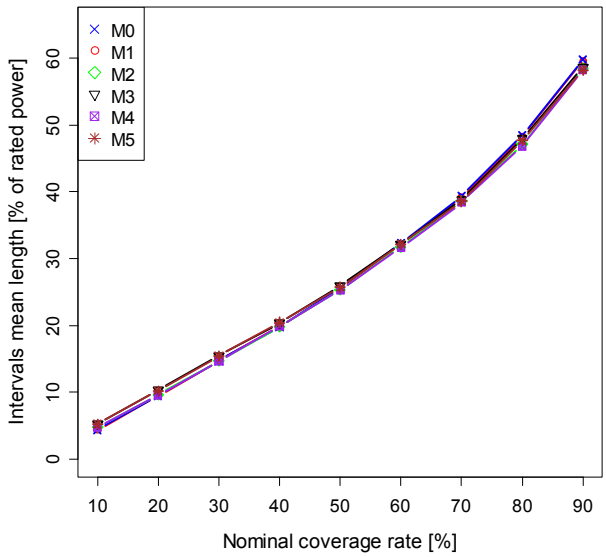

Fig. 3-231 Sharpness diagram for WFA with 6:00 AM NWP and splines QR models M0–M5 for look-ahead time step t+6h.

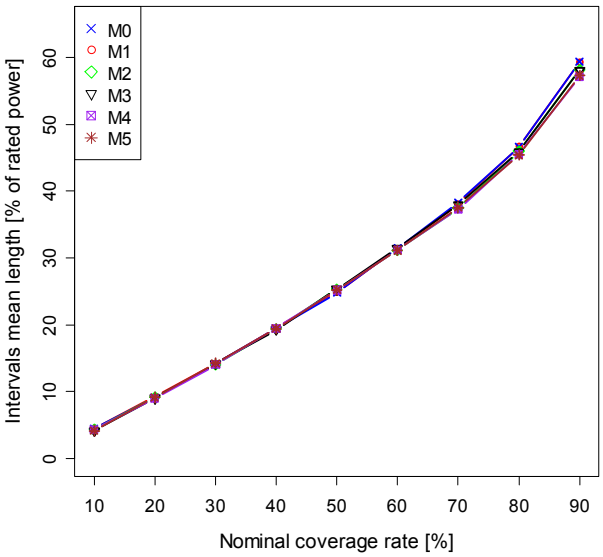

Fig. 3-232 Sharpness diagram for WFA with 6:00 PM NWP and splines QR models M0–M5 for look-ahead time step t+6h.

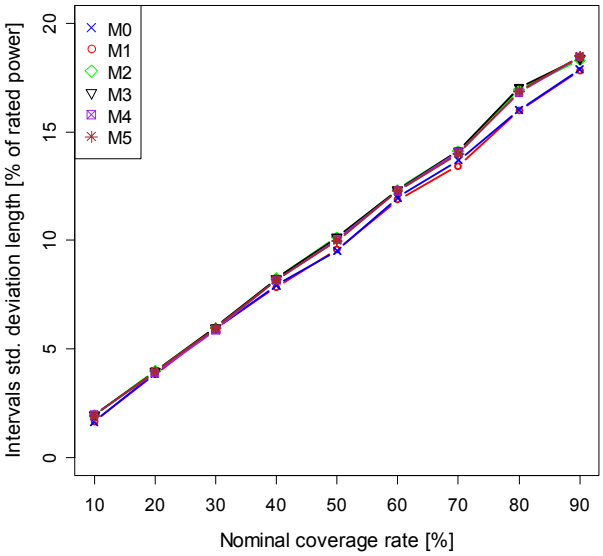

Fig. 3-233 Resolution diagram for WFA with 6:00 AM NWP and splines QR models M0–M5.

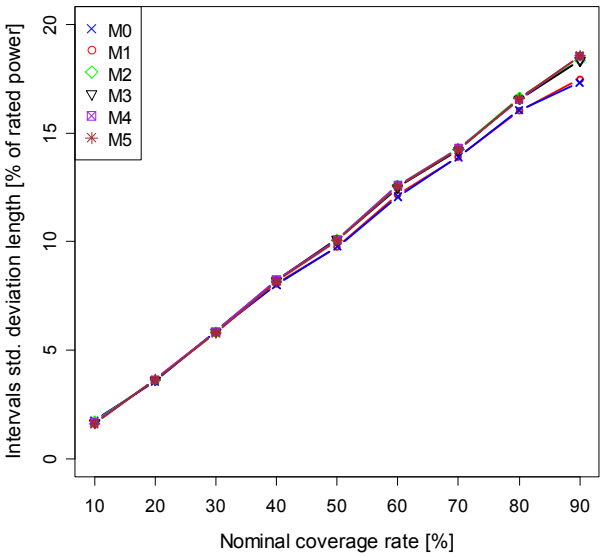

Fig. 3-234 Resolution diagram for WFA with 6:00 PM NWP and splines QR models M0–M5.

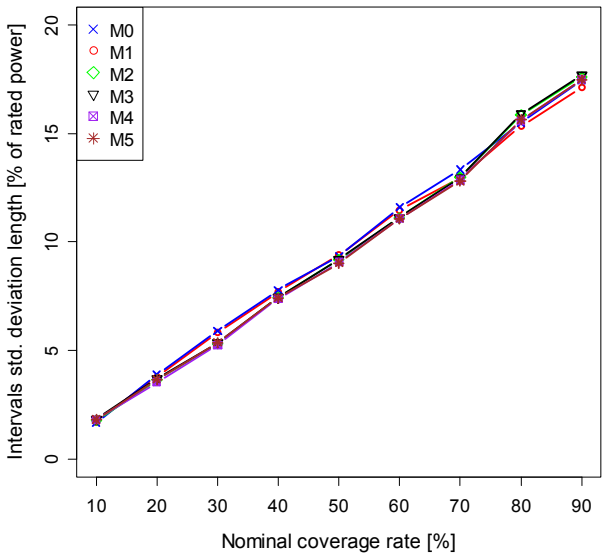

Fig. 3-235 Resolution diagram for WFA with 6:00 AM NWP and splines QR models M0–M5 for look-ahead time step t+6h.

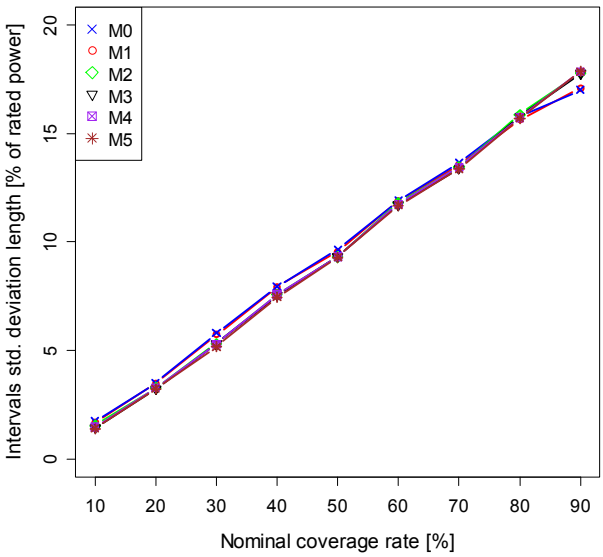

Fig. 3-236 Resolution diagram for WFA with 6:00 PM NWP and splines QR models M0–M5 for look-ahead time step t+6h.

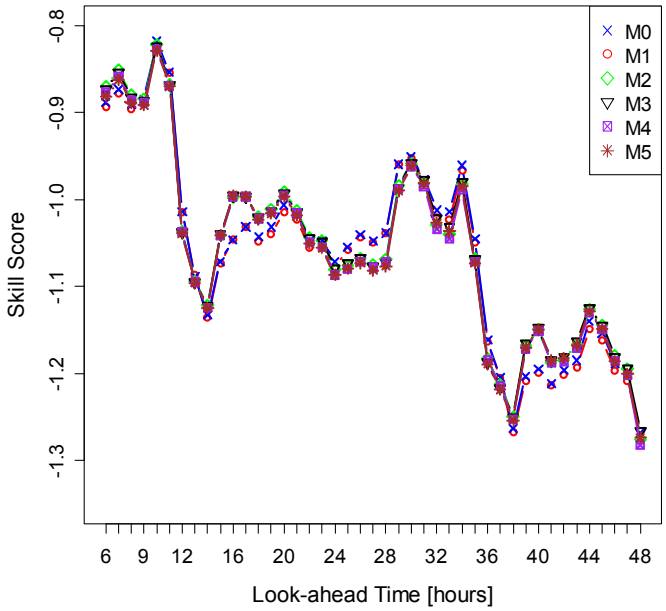

Fig. 3-237 Skill score diagram for WFA with 6:00 AM NWP and splines QR models M0–M5.

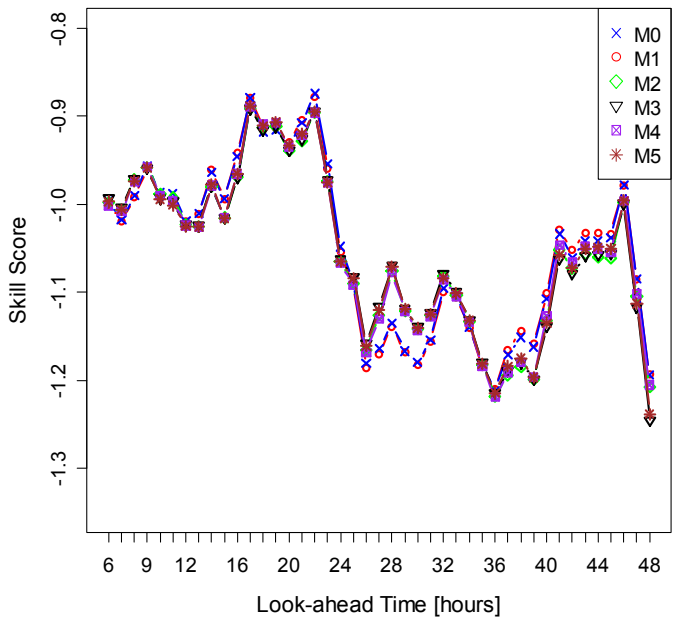

Fig. 3-238 Skill score diagram for WFA with 6:00 PM NWP and splines QR models M0–M5.

Concluding Results and Remarks

The performance of all models is very similar, and there is no model that dominates the others in all criteria. However, the look-ahead time step improves the calibration and skill score of the NW and QC estimators, whereas for the splines QR, a simple model with wind speed presents the best result. For some quantiles, the wind direction also improves the results.

Nevertheless, the model M3 with wind speed and look-ahead time step is the one that presents the best overall performance in the three uncertainty forecast algorithms. Hence, Fig. 3-239 through Fig. 3-246 resume the comparison between model M3 for the NW, QC, and splines QR estimators. Note that the calibration is presented for quantiles between 1% and 5% in 1% steps, then from 5% to 95% in 5% steps, and finally from 95% to 99% in 1% steps.

The following conclusions can be derived for wind farm A:

- NW estimator presents the best calibration performance for the 6 AM NWP;
- QC presents the best calibration for the 6 PM NWP;
- NW and QC have the same performance in terms of sharpness, resolution, and skill score;
- Splines QR presents the best sharpness and resolution performance;
- Splines QR presents the best calibration for the left tail, whereas KDF methods present the best calibration for the right tail;
- KDF methods have almost the same performance as QR in terms of skill score. Although QR is better than KDF for some look-ahead steps, it is also worse in others; and
- The methods with better calibration present a worse performance in terms of sharpness and resolution.

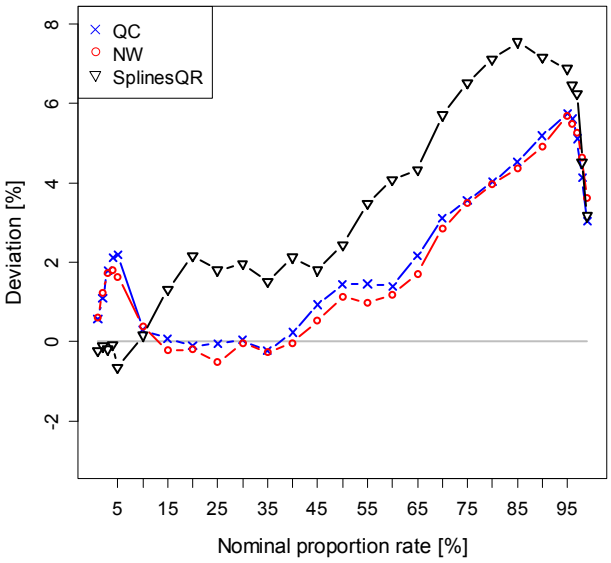

Fig. 3-239 Calibration diagram for WFA with 6:00 AM NWP and NW, QC, and QR models.

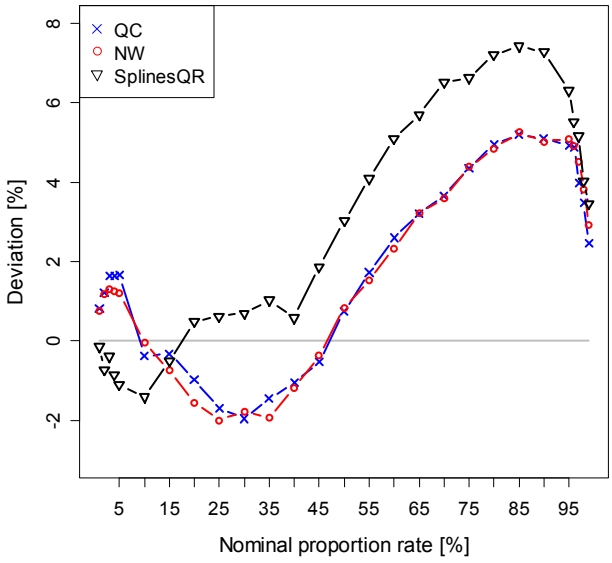

Fig. 3-240 Calibration diagram for WFA with 6:00 PM NWP and NW, QC, and QR models.

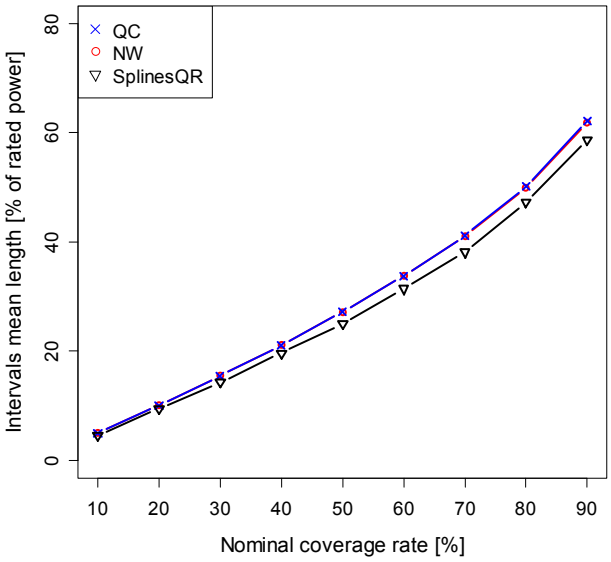

Fig. 3-241 Sharpness diagram for WFA with 6:00 AM NWP and NW, QC, and QR models.

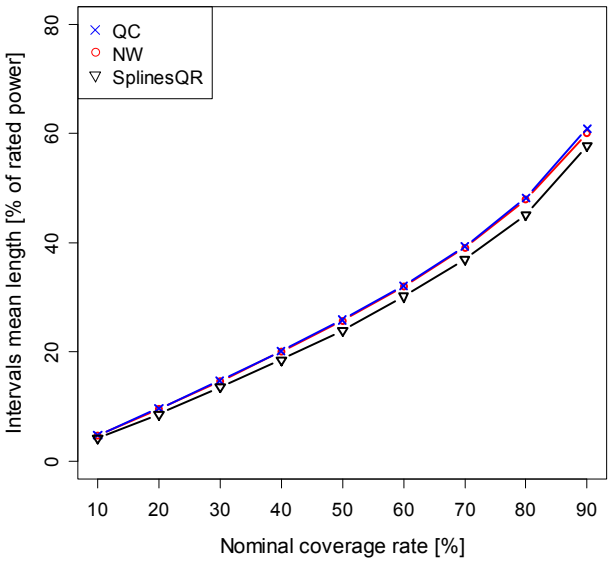

Fig. 3-242 Sharpness diagram for WFA with 6:00 PM NWP and NW, QC, and QR models.

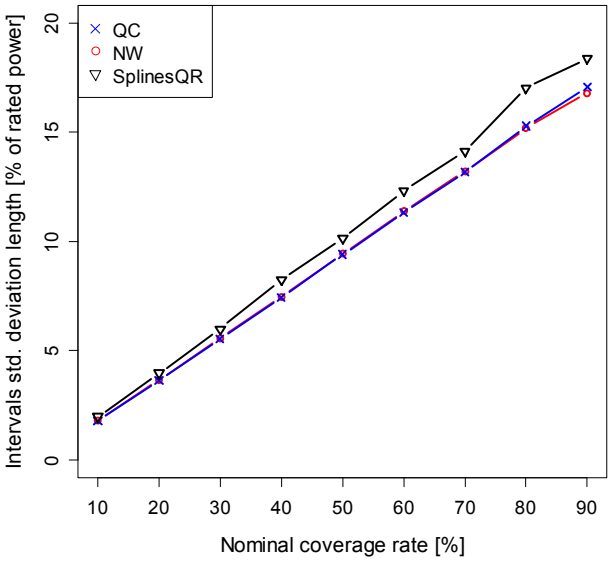

Fig. 3-243 Resolution diagram for WFA with 6:00 AM NWP and NW, QC, and QR models.

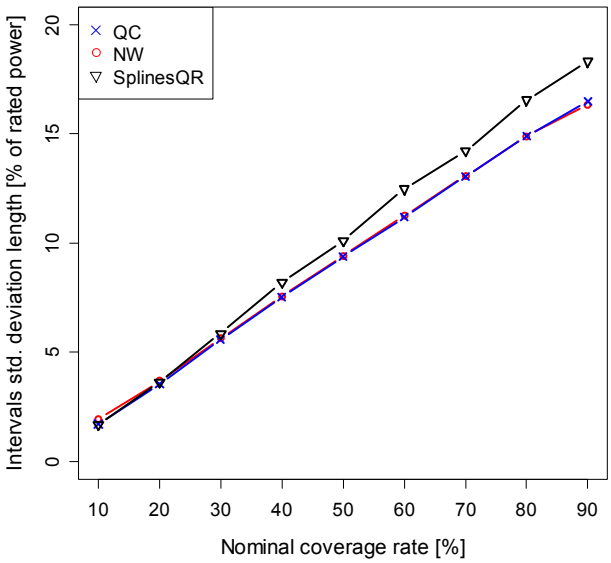

Fig. 3-244 Resolution diagram for WFA with 6:00 PM NWP and NW, QC, and QR models.

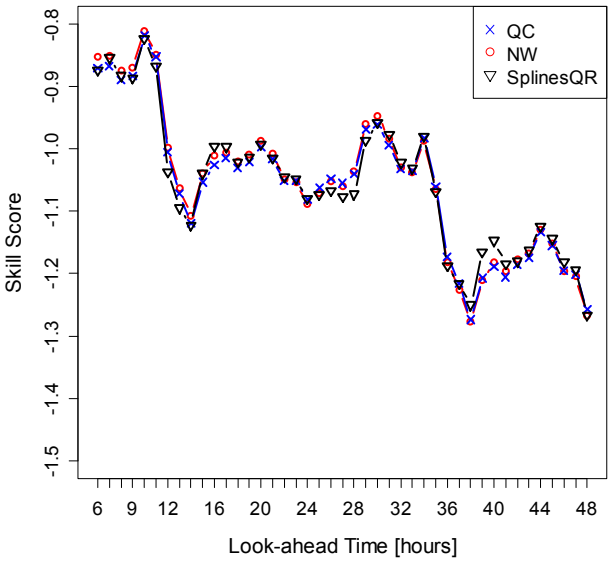

Fig. 3-245 Skill score diagram for WFA with 6:00 AM NWP and NW, QC, and QR models.

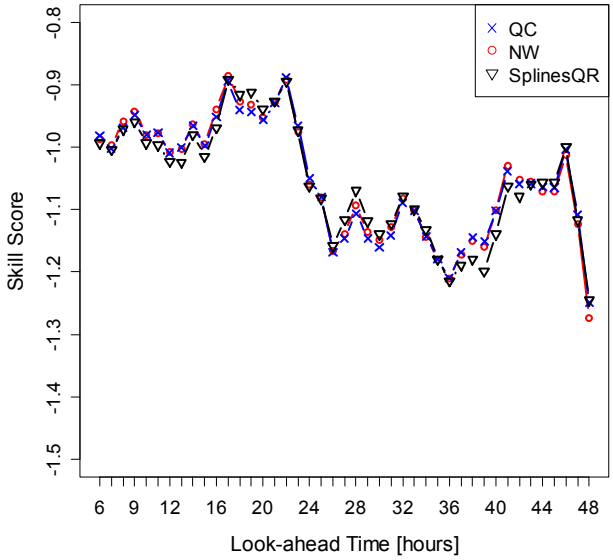

Fig. 3-246 Skill score diagram for WFA with 6:00 PM NWP and NW, QC, and QR models.

Wind Farm B

Nadaraya-Watson (NW) KDF

The following kernel functions were used in the NW estimator and WFB:

- Wind power generation: Chen's beta kernel from (3-17) with a bandwidth equal to 0.08;
- Wind speed forecast: Chen's gamma kernel from (3-21) with a bandwidth equal to 1.0;
- Wind direction: von Mises distribution from (3-24) with a bandwidth equal to 2.5;
- Look-ahead time step: Chen's beta kernel from (3-17) with a bandwidth equal to 0.1;
- Hour of the day: von Mises distribution from (3-24) with a bandwidth equal to 2.5.

The kernel bandwidth values were determined experimentally (via trial and error) and using as a starting point the values suggested by the function *cde.bandwidths* from the R package "hdrcde" [58].

Fig. 3-247 and Fig. 3-248 depict the calibration obtained for WFB using NWPs launched at 6 AM and 6 PM, respectively. The calibration performance is similar between the two figures. For quantiles below 55%, the models with the best performance are M0–M1 and M4–M5, whereas for quantiles above 55%, models M2–M3 present the best performance. In general, the models' performance is almost equal.

For this wind farm, the inclusion of the wind direction in the model helps improve the performance for some quantiles (see Fig. 3-248).

The calibration performance for look-ahead *t+6h* is depicted in Fig. 3-249 and Fig. 3-250, whereas the performance for *t+15h* can be found in Appendix C. When the analysis is performed for each look-ahead time, there are more variations between models. Although it is difficult to identify the best models in these figures, it seems that the model that only used wind speed (M0) presents the worst performance.

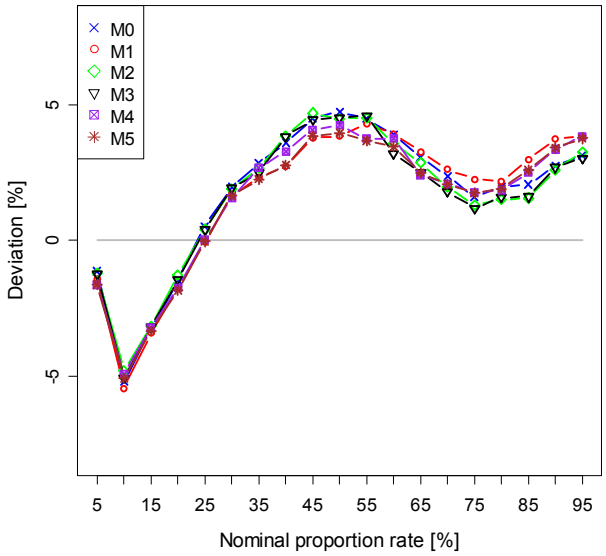

Fig. 3-247 Calibration diagram for WFB with 6:00 AM NWP and NW models M0–M5.

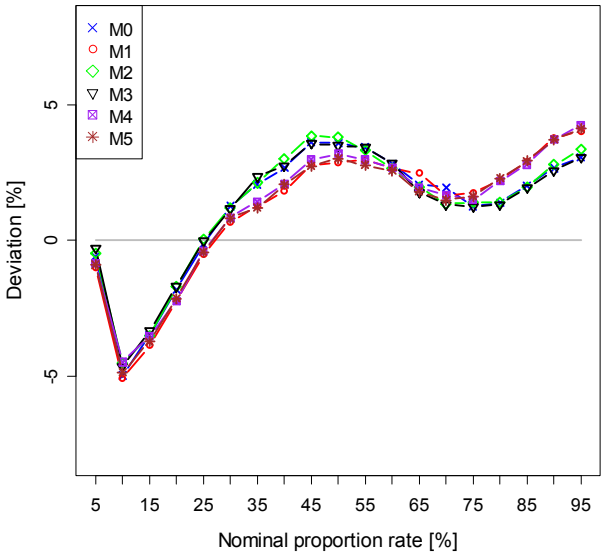

Fig. 3-248 Calibration diagram for WFB with 6:00 PM NWP and NW models M0–M5.

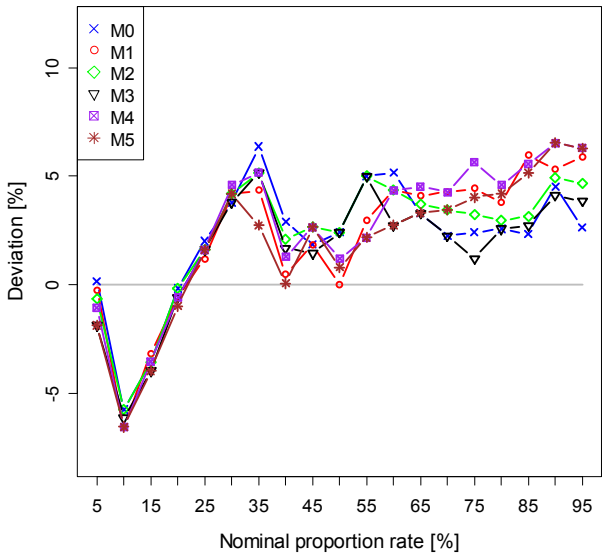

Fig. 3-249 Calibration diagram for WFB with 6:00 AM NWP and NW models M0–M5 for look-ahead time step t+6h.

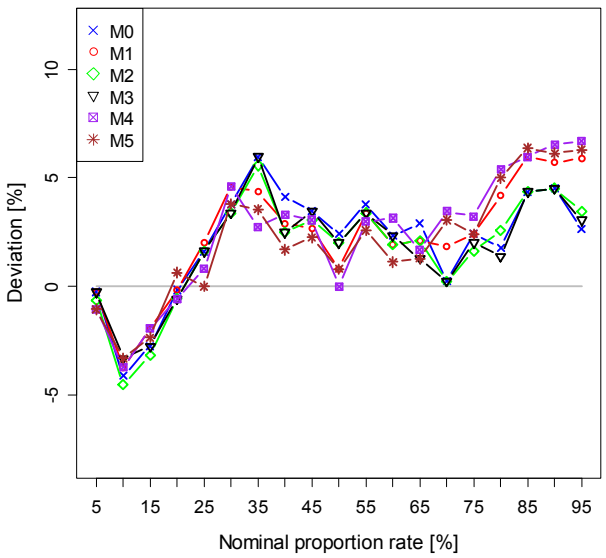

Fig. 3-250 Calibration diagram for WFB with 6:00 PM NWP and NW models M0–M5 for look-ahead time step t+6h.

Fig. 3-251 and Fig. 3-252 depict the sharpness obtained for WFB with 6 AM and 6 PM NWPs. The sharpness is rather similar for all models; the same is detected when it is computed for a specific look-ahead time step, such as *t+6h* in Fig. 3-253 and Fig. 3-254, or *t+15h* in Appendix C. Nevertheless, models M4 and M5 present the best sharpness performance.

Fig. 3-255 and Fig. 3-256 depict the resolution obtained for WFB. In this case, the model M4 presents the best performance for both the 6 AM and 6 PM NWPs. The worst performance is from M0. When the analysis is performed for look-ahead time step *t+6h* (Fig. 3-257 and Fig. 3-258), the model with the best performance is M1; however, for *t+15h* (see Appendix C), the best performance is from M4.

Fig. 3-259 and Fig. 3-260 present the skill score computed for each look-ahead time step for the 6 AM and 6 PM NWPs. The skill score performance is not significantly different for all models. The models with the best performance are M2 and M3 for both 6 AM and 6 PM. The worst performance is from model M1.

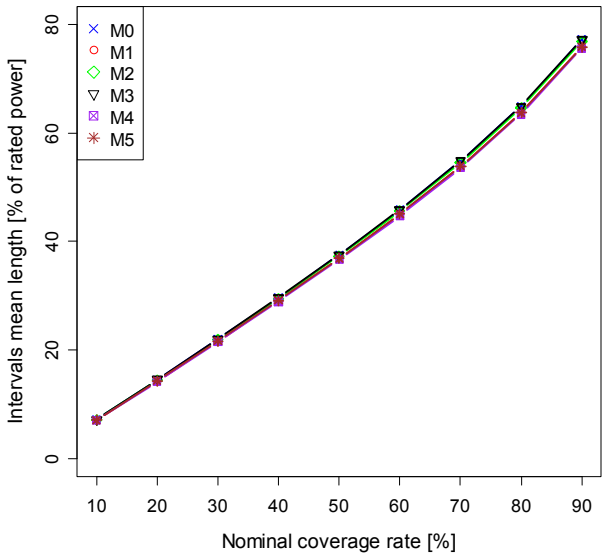

Fig. 3-251 Sharpness diagram for WFB with 6:00 AM NWP and NW models M0–M5.

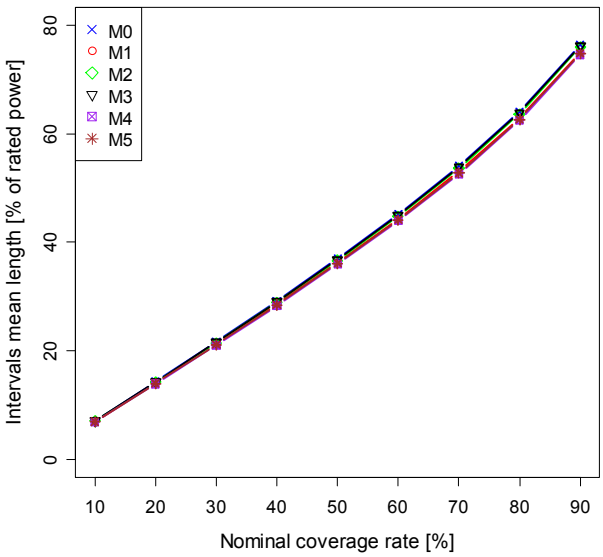

Fig. 3-252 Sharpness diagram for WFB with 6:00 PM NWP and NW models M0–M5.

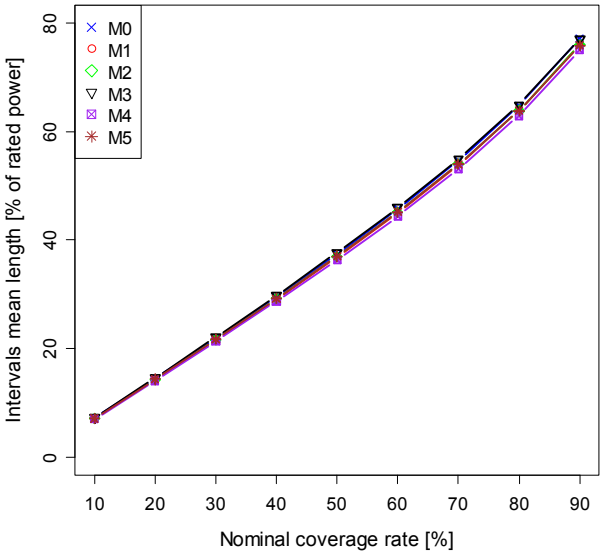

Fig. 3-253 Sharpness diagram for WFA with 6:00 AM NWP and NW models M0–M5 for look-ahead time step t+6h.

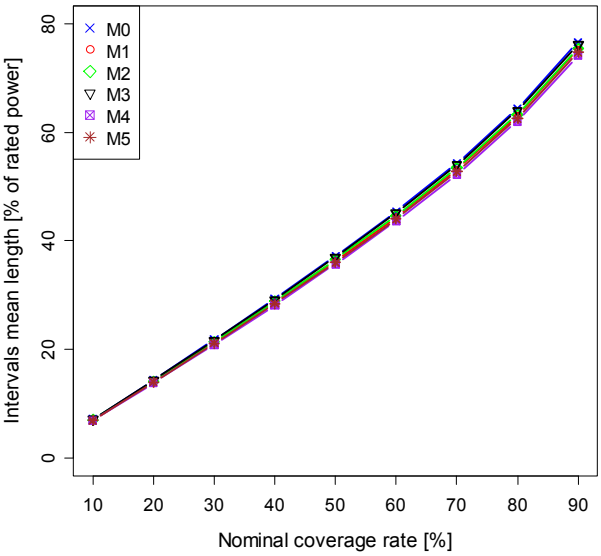

Fig. 3-254 Sharpness diagram for WFA with 6:00 PM NWP and NW models M0–M5 for look-ahead time step t+6h.

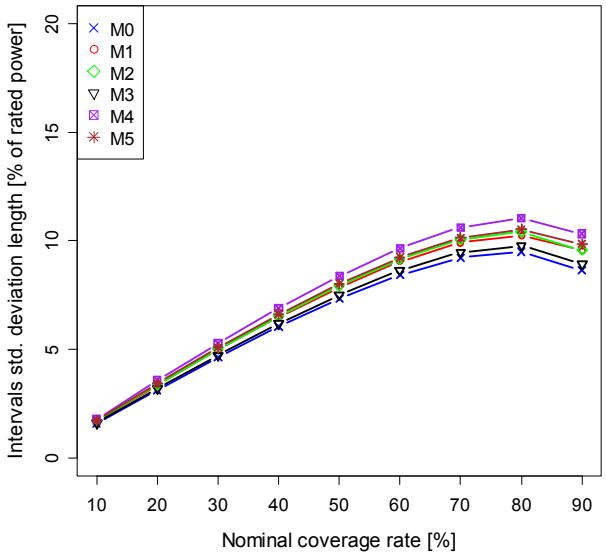

Fig. 3-255 Resolution diagram for WFB with 6:00 AM NWP and NW models M0–M5.

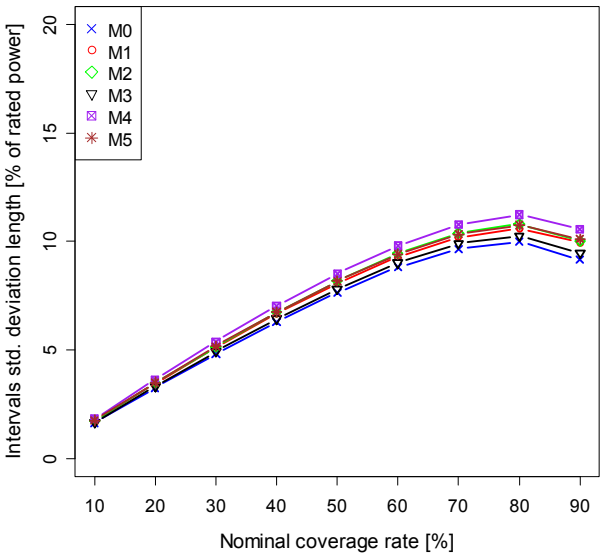

Fig. 3-256 Resolution diagram for WFB with 6:00 PM NWP and NW models M0–M5.

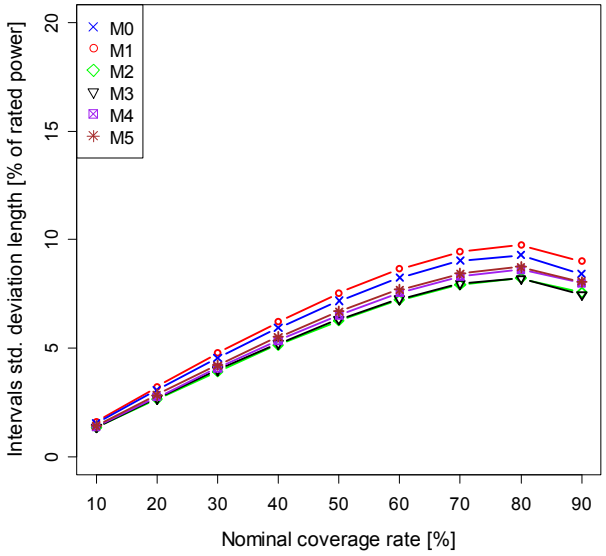

Fig. 3-257 Resolution diagram for WFB with 6:00 AM NWP and NW models M0–M5 for look-ahead time step t+6h.

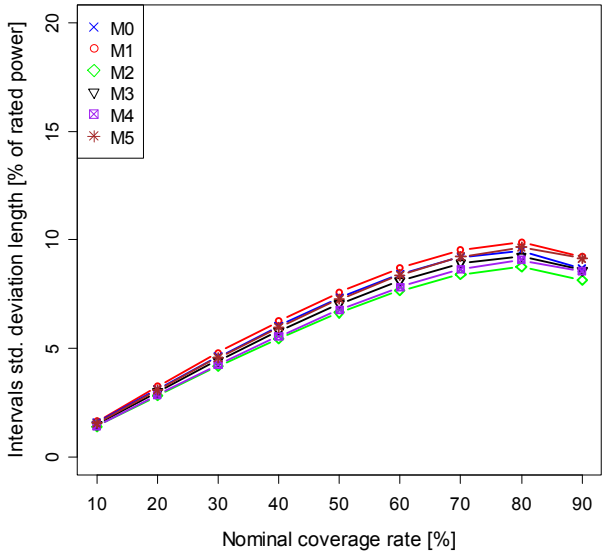

Fig. 3-258 Resolution diagram for WFB with 6:00 PM NWP and NW models M0–M5 for look-ahead time step t+6h.

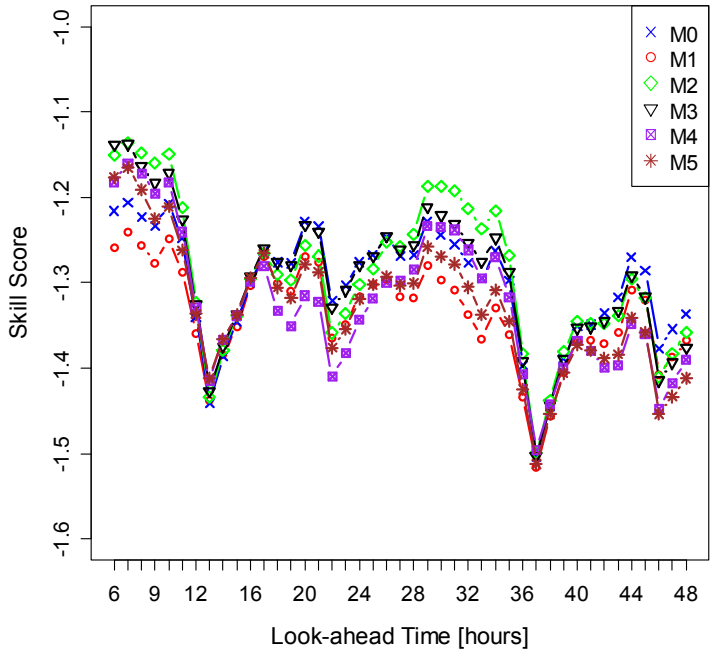

Fig. 3-259 Skill score diagram for WFB with 6:00 AM NWP and NW models M0–M5.

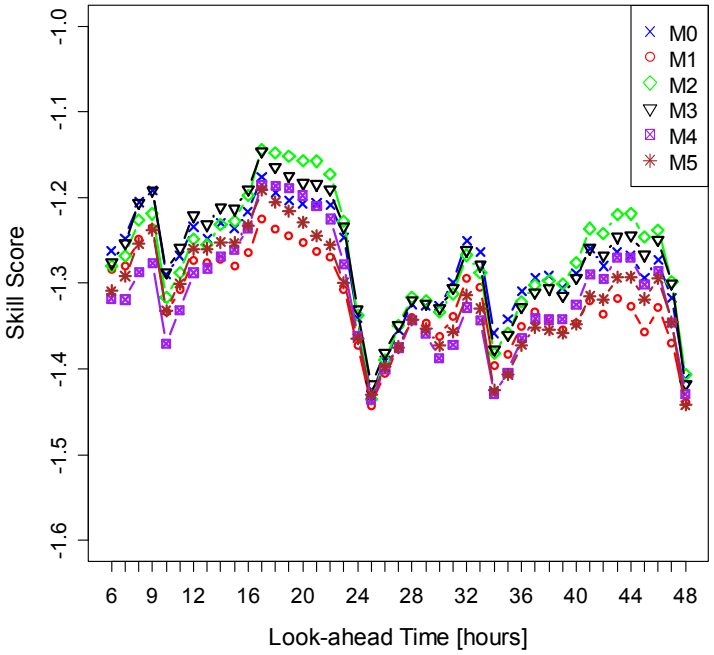

Fig. 3-260 Skill score diagram for WFB with 6:00 PM NWP and NW models M0–M5.

Quantile-Copula (QC) KDF

The following kernel functions were used in the QC estimator and WFA:

- Wind power generation: Chen's beta kernel from (3-17) with a bandwidth equal to 0.04;
- Wind speed forecast: Chen's beta kernel from (3-17) with a bandwidth equal to 0.04;
- Wind direction: von Mises distribution from (3-24) with a bandwidth equal to 1.0;
- Look-ahead time step: Chen's beta kernel from (3-17) with a bandwidth equal to 0.2;
- Hour of the day: von Mises distribution from (3-24) with a bandwidth equal to 1.0.

The kernel bandwidth values were determined experimentally (via trial and error) and using as a starting point the values suggested by the function *cde.bandwidths* from the R package "hdrcde" [58].

Fig. 3-261 and Fig. 3-262 depict the calibration for WFB using NWPs launched at 6 AM and 6 PM, respectively. The best performance is from models M2 and M3. Nevertheless, there is no significant difference between models' performance.

Fig. 3-263 and Fig. 3-264 depict calibration diagrams for look-ahead time step *t+6h* obtained with 6 AM and 6 PM NWPs. The calibration for *t+15h* can be found in Appendix C. From these figures, it is difficult to distinguish the model with the best performance; however, model M3 presents a regular performance.

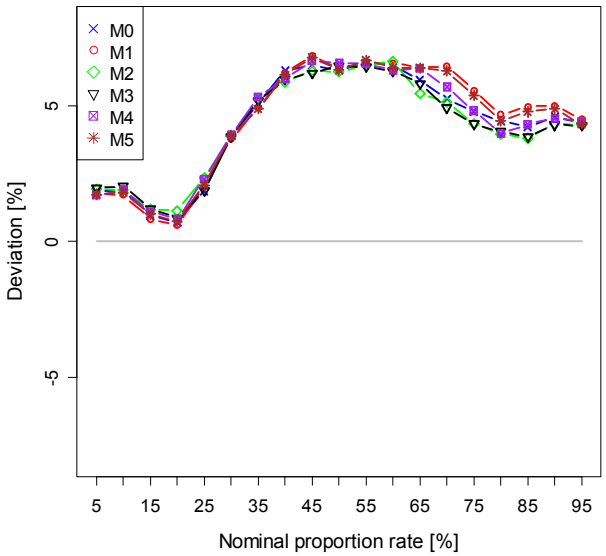

Fig. 3-261 Calibration diagram for WFB with 6:00 AM NWP and NW models M0–M5.

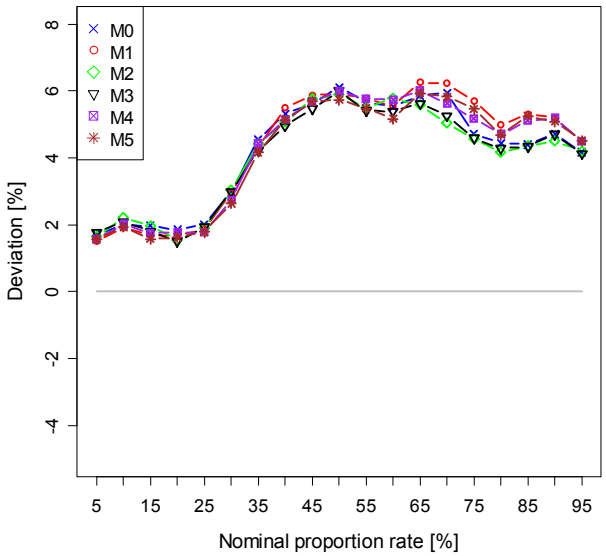

Fig. 3-262 Calibration diagram for WFB with 6:00 PM NWP and NW models M0–M5.

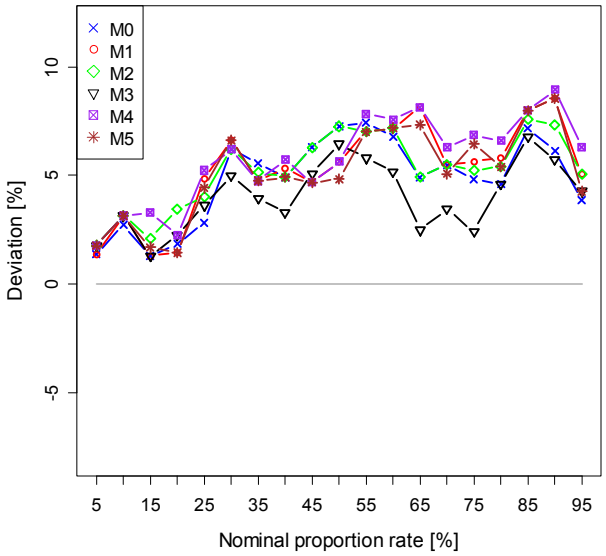

Fig. 3-263 Calibration diagram for WFB with 6:00 AM NWP and QC models M0–M5 for look-ahead time step t+6h.

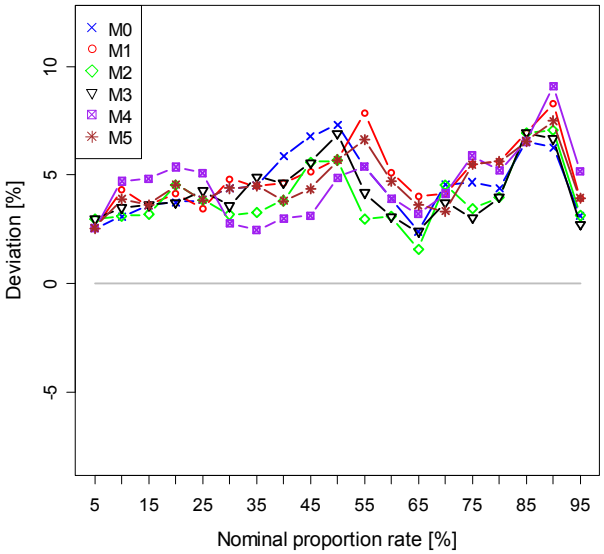

Fig. 3-264 Calibration diagram for WFB with 6:00 PM NWP and QC models M0–M5 for look-ahead time step t+6h.

Fig. 3-265 and Fig. 3-266 depict the sharpness obtained for WFB with 6 AM and 6 PM NWPs. The sharpness is almost the same for all models; the same is verified when it is computed for a specific look-ahead time step, such as *t+6h* in Fig. 3-267 and Fig. 3-268, or *t+15h* in Appendix C.

Fig. 3-269 and Fig. 3-270 depict the resolution obtained for WFB. The models' performance is almost similar, but with a slight advantage for models M4 and M2 for both the 6 AM and 6 PM NWPs. When the analysis is performed for look-ahead time step *t+6h* (Fig. 3-271 and Fig. 3-272), the differences are more distinct, but models M4 and M2 also present the best performance.

Fig. 3-273 and Fig. 3-274 present the skill score computed for each look-ahead time step for the 6 AM and 6 PM NWPs. The performance of all of these models is rather similar, with a slight advantage for models M2 and M3 at both 6 AM and 6 PM.

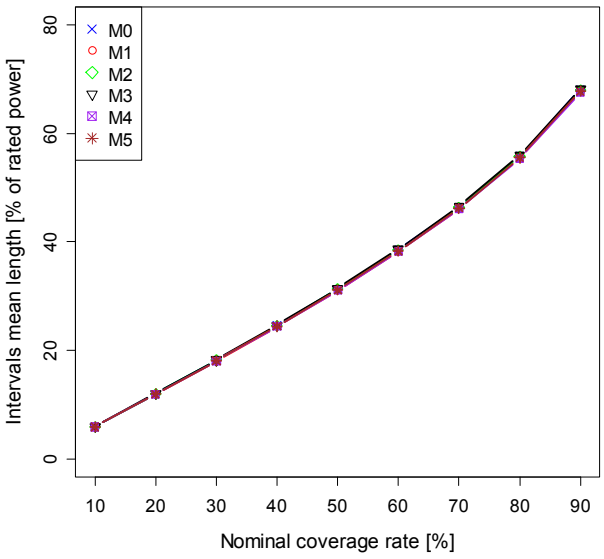

Fig. 3-265 Sharpness diagram for WFB with 6:00 AM NWP and QC models M0–M5.

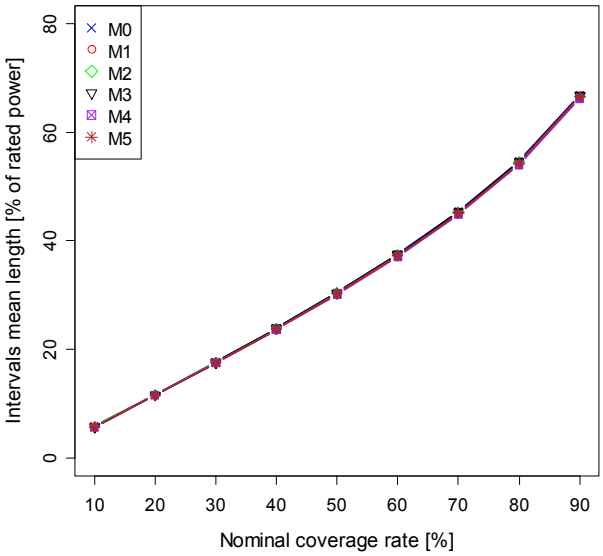

Fig. 3-266 Sharpness diagram for WFB with 6:00 PM NWP and QC models M0–M5.

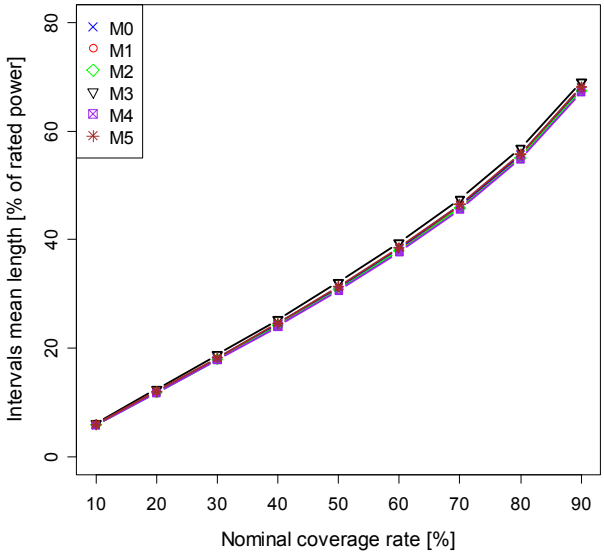

Fig. 3-267 Sharpness diagram for WFA with 6:00 AM NWP and QC models M0–M5 for look-ahead time step t+6h.

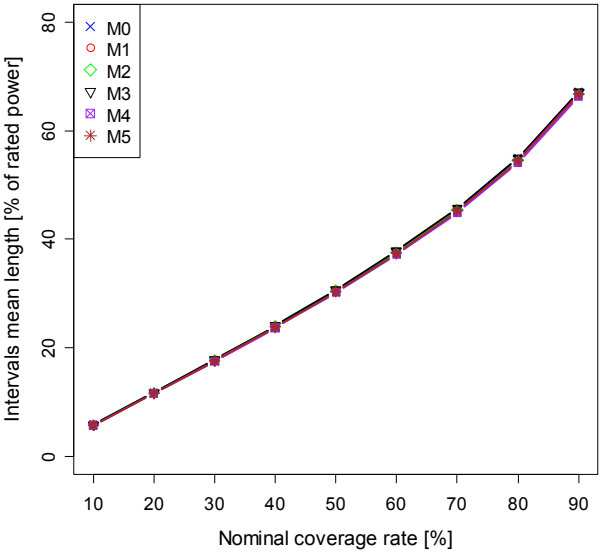

Fig. 3-268 Sharpness diagram for WFA with 6:00 PM NWP and QC models M0–M5 for look-ahead time step t+6h.

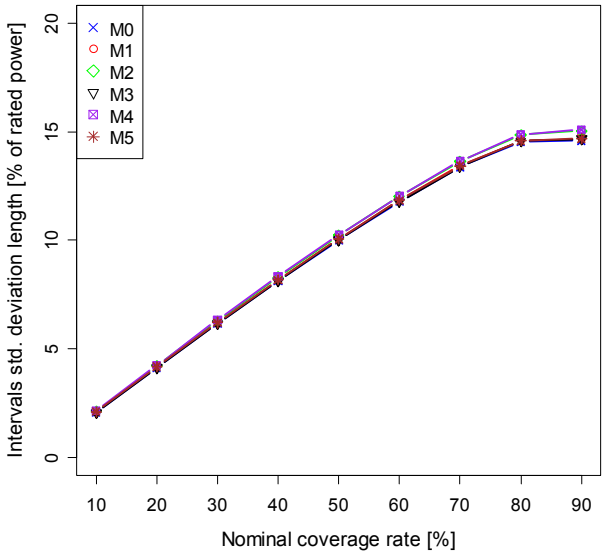

Fig. 3-269 Resolution diagram for WFB with 6:00 AM NWP and QC models M0–M5.

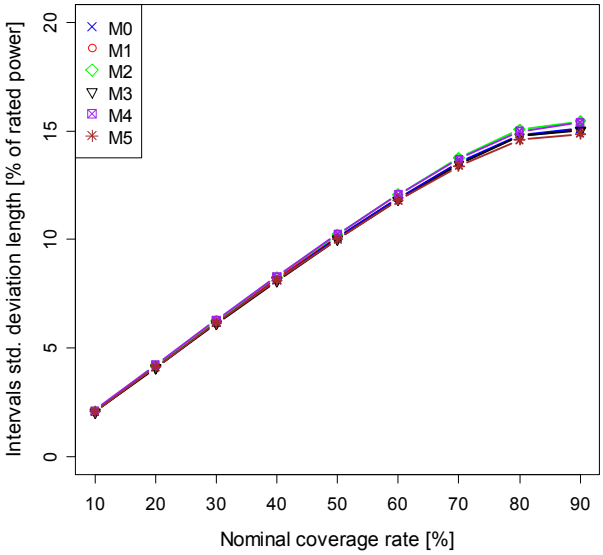

Fig. 3-270 Resolution diagram for WFB with 6:00 PM NWP and QC models M0–M5.

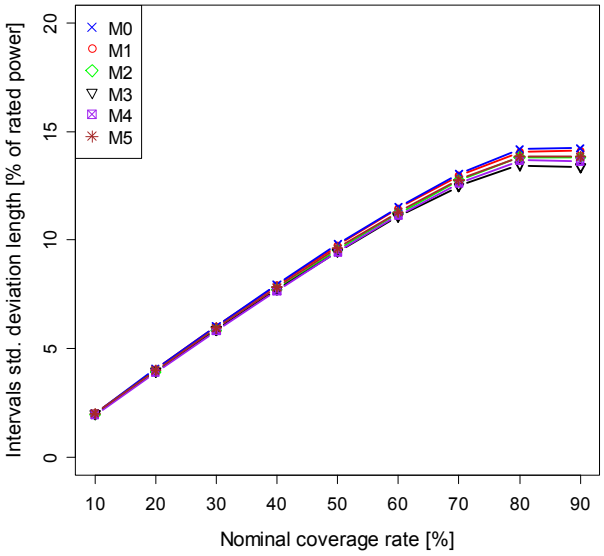

Fig. 3-271 Resolution diagram for WFB with 6:00 AM NWP and QC models M0–M5 for look-ahead time step t+6h.

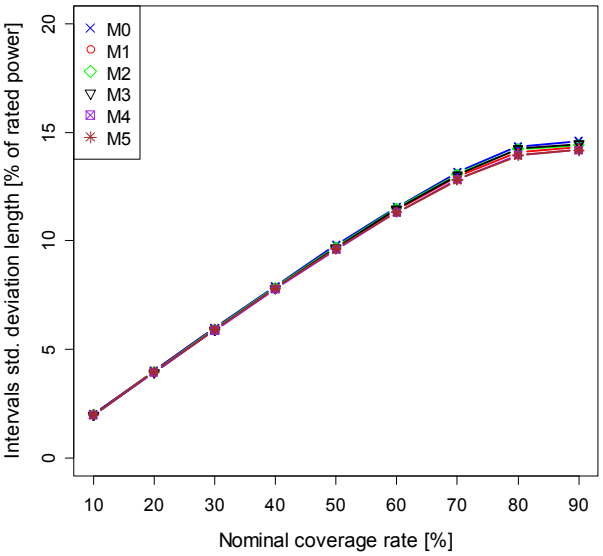

Fig. 3-272 Resolution diagram for WFB with 6:00 PM NWP and QC models M0–M5 for look-ahead time step t+6h.

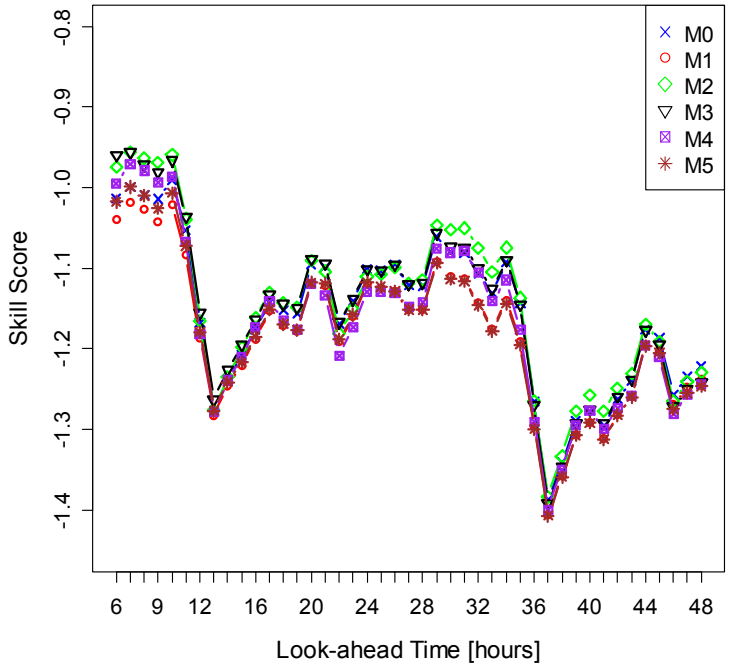

Fig. 3-273 Skill score diagram for WFB with 6:00 AM NWP and QC models M0–M5.

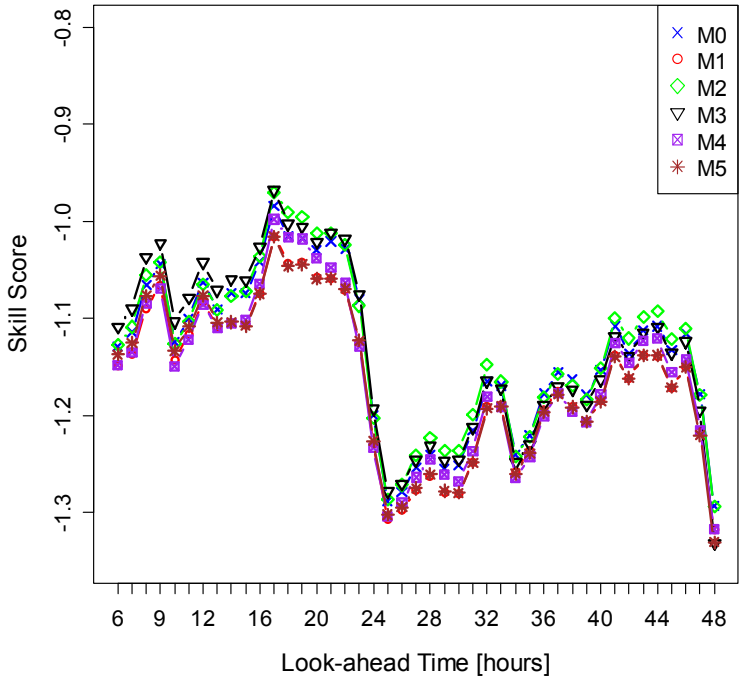

Fig. 3-274 Skill score diagram for WFB with 6:00 PM NWP and QC models M0–M5.

Splines Quantile Regression (splines QR)

Fig. 3-275 and Fig. 3-276 depict the calibration obtained for WFB and using NWP launched at 6 AM and 6 PM, respectively. The calibration between the two figures is almost equal. The models with the best calibration performance are M0, M2, and M3; however, the overall performance is generally very similar. The inclusion of wind direction reduces the calibration performance for some quantiles.

Fig. 3-277 and Fig. 3-278 depict the calibration diagram for look-ahead time step *t+6h* obtained with 6 AM and 6 PM NWPs. For this look-ahead time step, it is very difficult to find the model with the best performance.

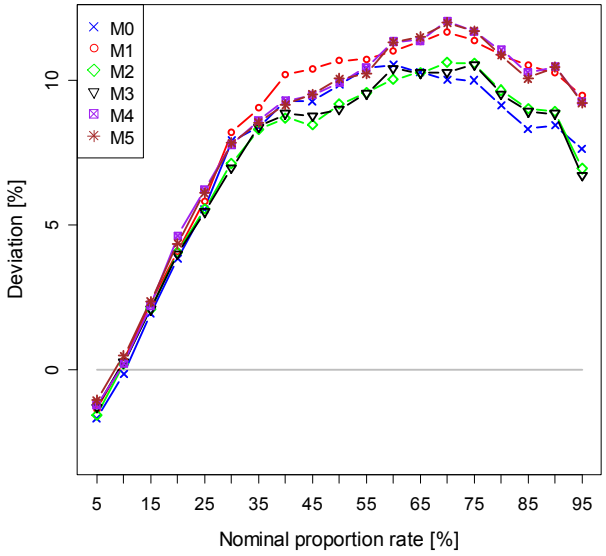

Fig. 3-275 Calibration diagram for WFB with 6:00 AM NWP and splines QR models M0–M5.

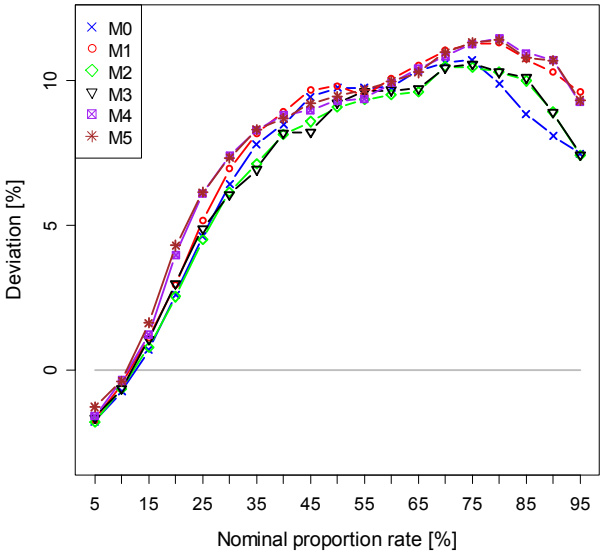

Fig. 3-276 Calibration diagram for WFB with 6:00 PM NWP and splines QR models M0–M5.

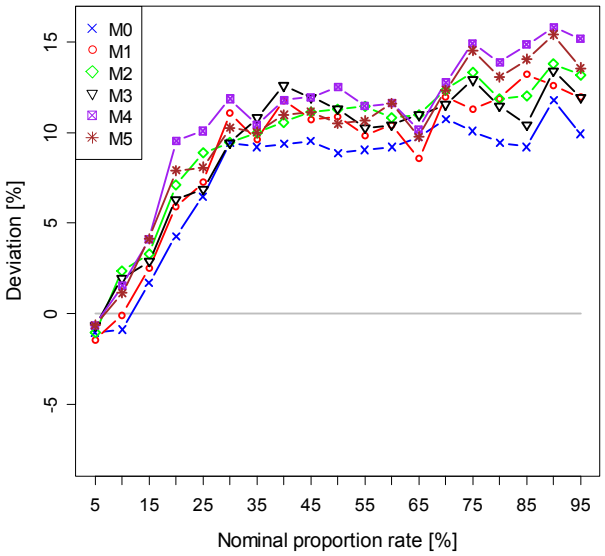

Fig. 3-277 Calibration diagram for WFB with 6:00 AM NWP and splines QR models M0–M5 for look-ahead time step t+6h.

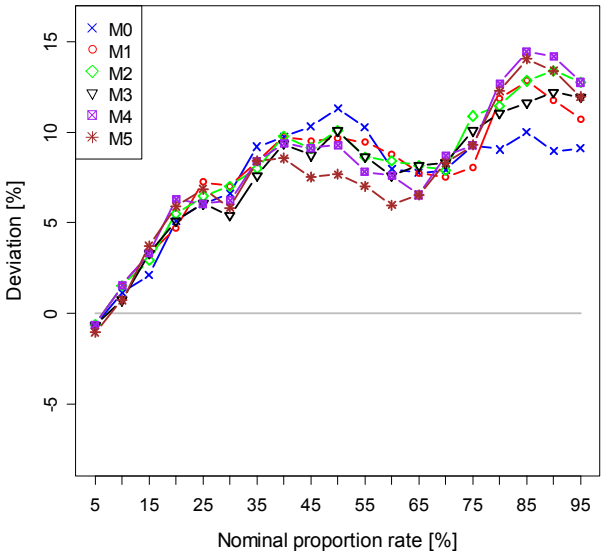

Fig. 3-278 Calibration diagram for WFB with 6:00 PM NWP and splines QR models M0–M5 for look-ahead time step t+6h.

Fig. 3-279 and Fig. 3-280 depict the sharpness obtained for WFB with 6 AM and 6 PM NWPs. The sharpness is almost equal for all models, with a slight advantage for model M0. The same is detected for look-ahead time step *t+6h* (Fig. 3-281 and Fig. 3-282).

Fig. 3-283 and Fig. 3-284 depict the resolution obtained for WFB. In this case, the models from M2 and M3 present the best performance for both the 6 AM and 6 PM NWPs, and also for look-ahead time step *t+6h* (depicted in Fig. 3-285 and Fig. 3-286).

Fig. 3-287 and Fig. 3-288 present the skill score computed for each look-ahead time step and for the 6 AM and 6 PM NWPs. The models with the best performance are M2–M5, whereas models M0 (only for 6 AM) and M1 present the worst performance.

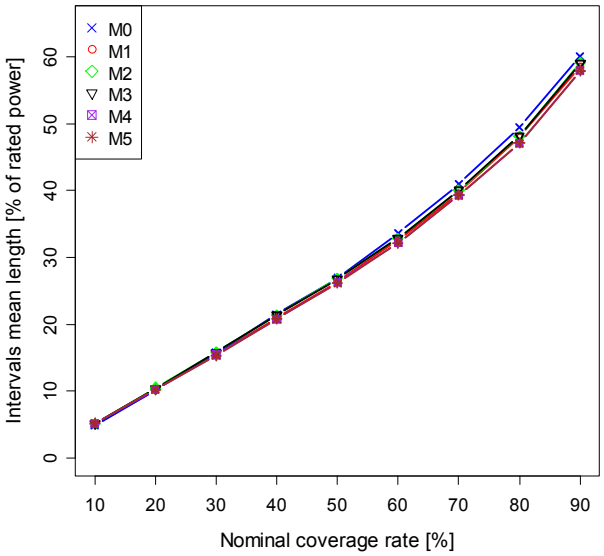

Fig. 3-279 Sharpness diagram for WFB with 6:00 AM NWP and splines QR models M0–M5.

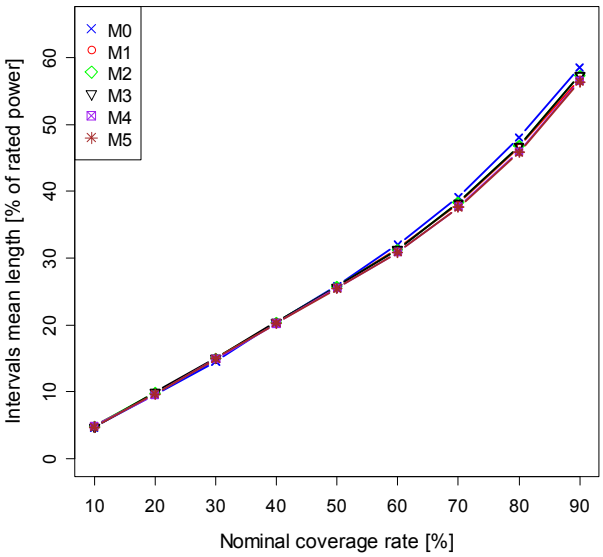

Fig. 3-280 Sharpness diagram for WFB with 6:00 PM NWP and splines QR models M0–M5.

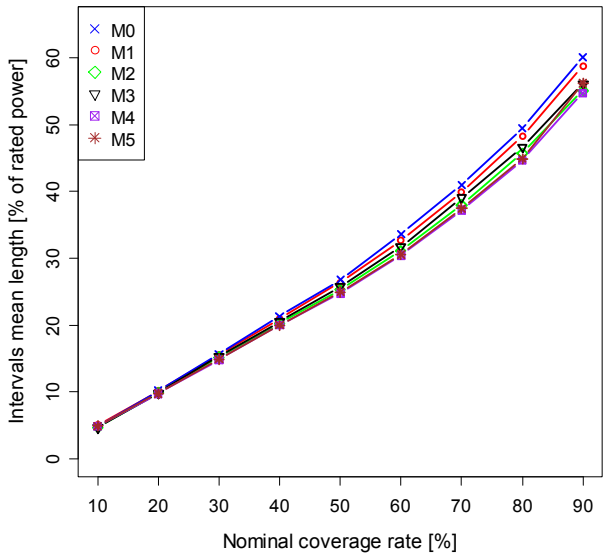

Fig. 3-281 Sharpness diagram for WFB with 6:00 AM NWP and splines QR models M0–M5 for look-ahead time step t+6h.

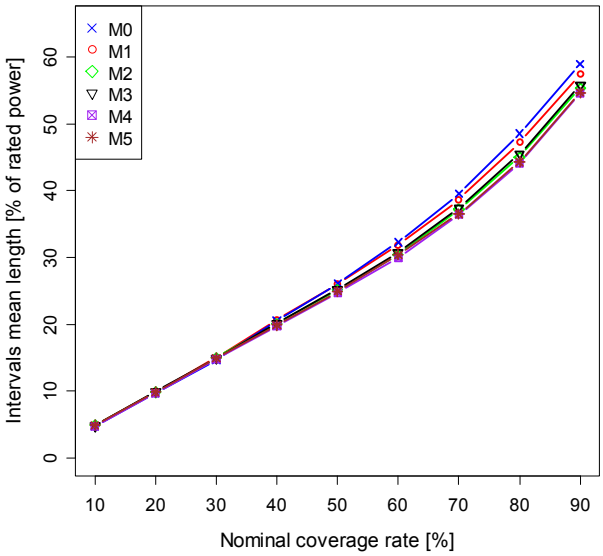

Fig. 3-282 Sharpness diagram for WFB with 6:00 PM NWP and splines QR models M0–M5 for look-ahead time step t+6h.

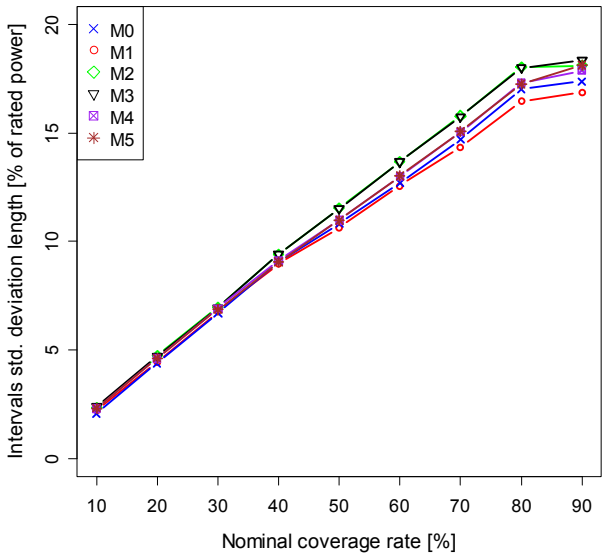

Fig. 3-283 Resolution diagram for WFB with 6:00 AM NWP and splines QR models M0–M5.

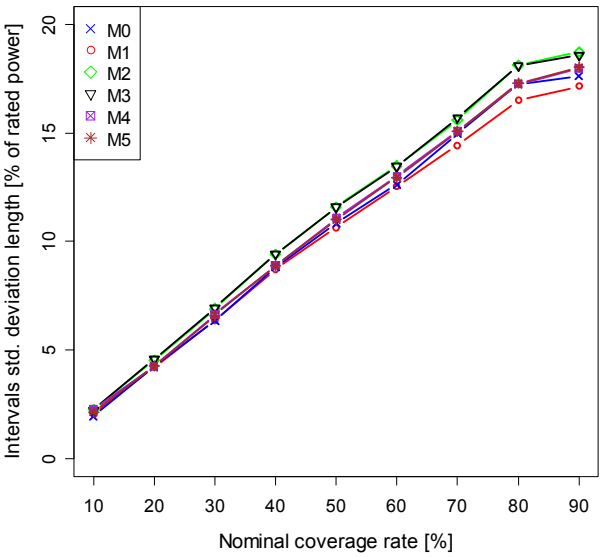

Fig. 3-284 Resolution diagram for WFB with 6:00 PM NWP and splines QR models M0–M5.

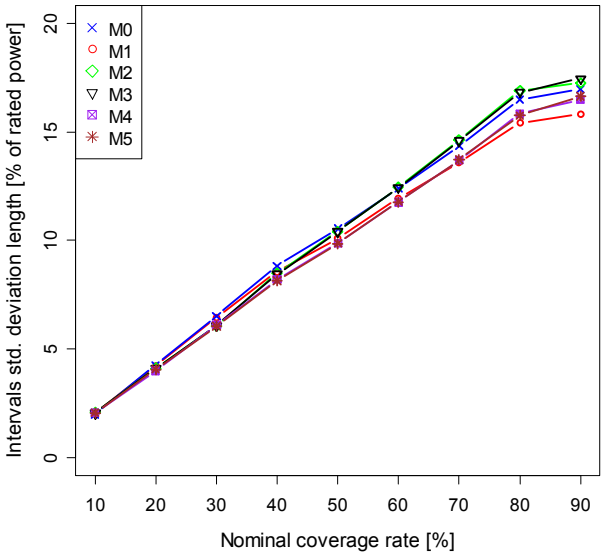

Fig. 3-285 Resolution diagram for WFB with 6:00 AM NWP and splines QR models M0–M5 for look-ahead time step t+6h.

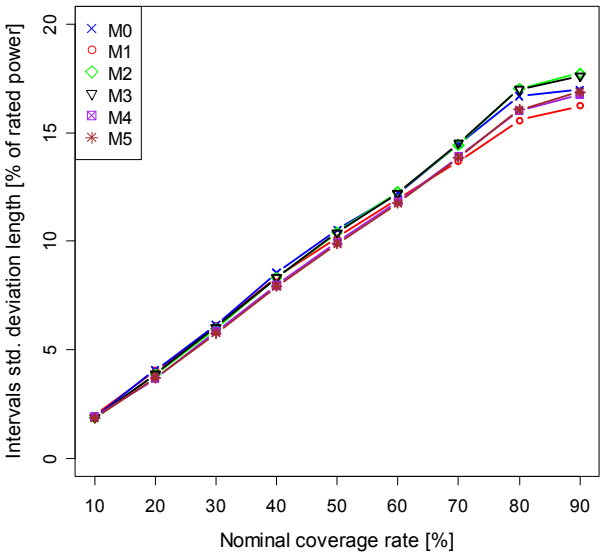

Fig. 3-286 Resolution diagram for WFB with 6:00 PM NWP and splines QR models M0–M5 for look-ahead time step t+6h.

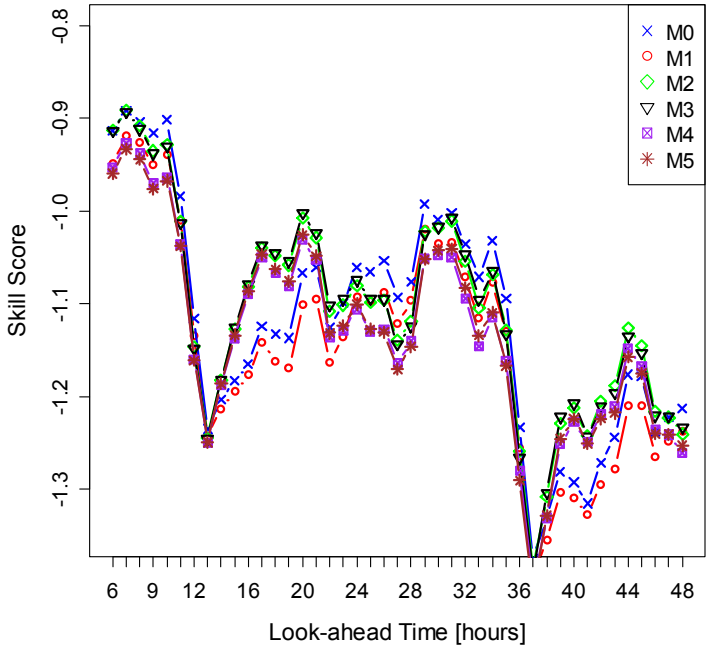

Fig. 3-287 Skill score diagram for WFB with 6:00 AM NWP and splines QR models M0–M5.

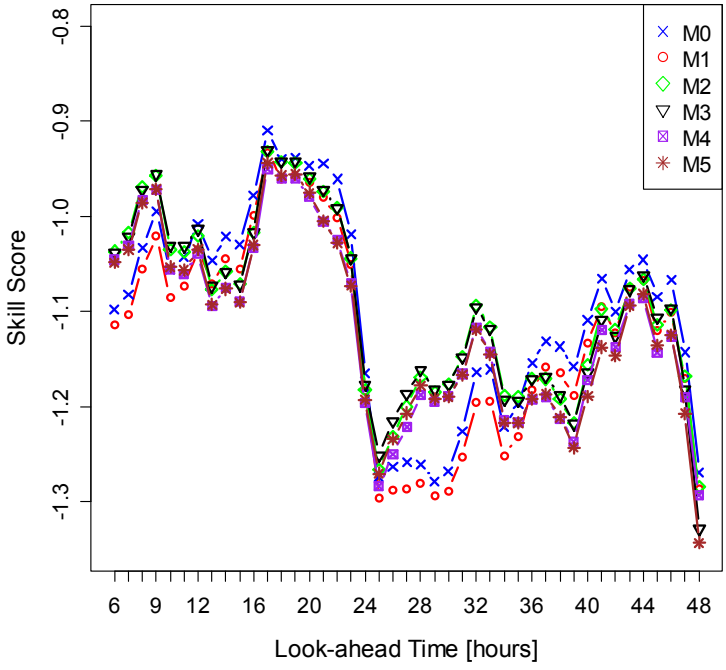

Fig. 3-288 Skill score diagram for WFB with 6:00 PM NWP and splines QR models M0–M5.

Concluding Results and Remarks

In the case of this wind farm, the performance of all models is also very similar, and there is no model that dominates the others in all criteria. The model M3 with wind speed and the look-ahead time step was the one that presents the best overall performance in the three uncertainty forecast algorithms. Hence, Fig. 3-289 through Fig. 3-296 resume the comparison between model M3 for the NW, QC, and splines QR estimators. Note that the calibration is presented for quantiles between 1% and 5% in 1% steps, then from 5% to 95% in 5% steps, and finally from 95% to 99% in 1% steps.

The following conclusions can be derived for wind farm B:

- The NW estimator presents the best overall calibration performance;
- QC presents good calibration performance;
- QC presents a better performance in terms of sharpness and resolution when compared to NW;
- Splines QR presents the best sharpness and resolution performance;
- The QR approaches underestimate the quantiles of the left tail, whereas the KDF methods overestimate them;
- The QR estimators present the best performance for the right tail;
- Splines QR presents the best results in terms of skill score; Pinson *et al.* [54] mentioned that the skill score of (3-44) is a generalization of the loss function considered in quantile regression, hence, it could justify why quantile regression presents the best performance in this criteria;
- NW presents a significantly worse performance in terms of skill score. This result was attributable to a worse performance in terms of sharpness and calibration; and
- QC presents a competitive performance with QR in terms of the skill score.

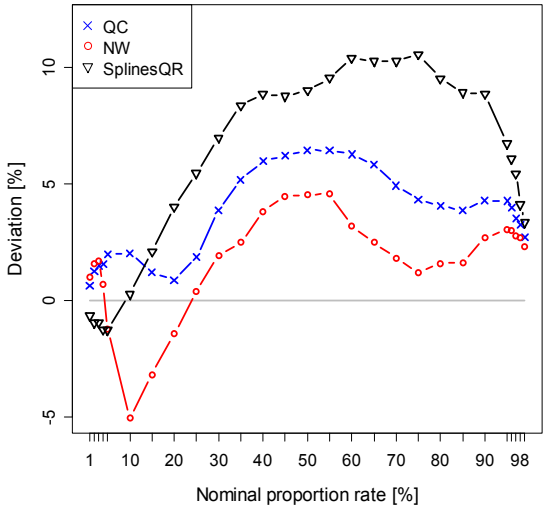

Fig. 3-289 Calibration diagram for WFB with 6:00 AM NWP and NW, QC, and QR models.

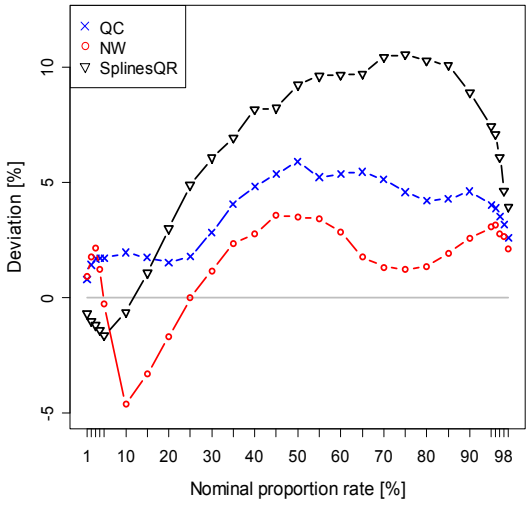

Fig. 3-290 Calibration diagram for WFB with 6:00 PM NWP and NW, QC, and QR models.

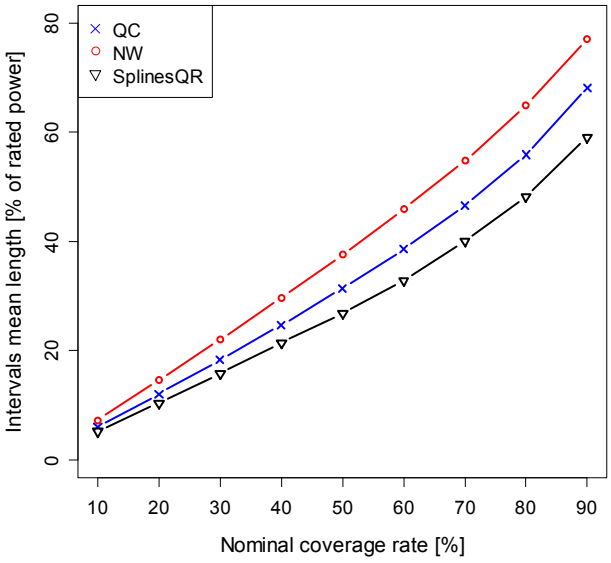

Fig. 3-291 Sharpness diagram for WFB with 6:00 AM NWP and NW, QC, and QR models.

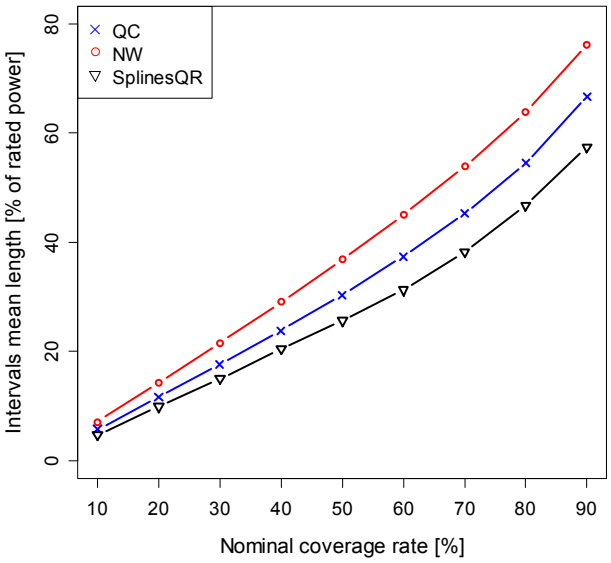

Fig. 3-292 Sharpness diagram for WFB with 6:00 PM NWP and NW, QC, and QR models.

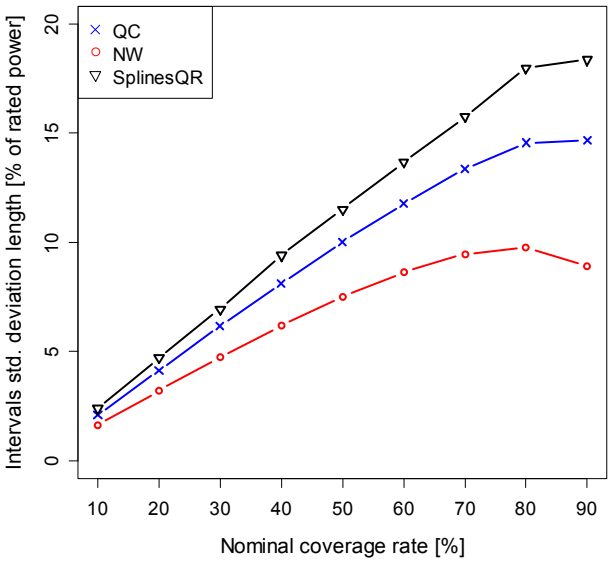

Fig. 3-293 Resolution diagram for WFB with 6:00 AM NWP and NW, QC, and QR models.

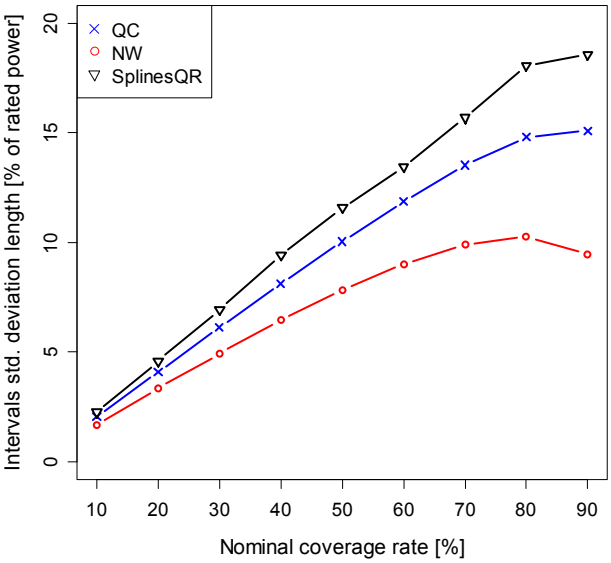

Fig. 3-294 Resolution diagram for WFB with 6:00 PM NWP and NW, QC, and QR models.

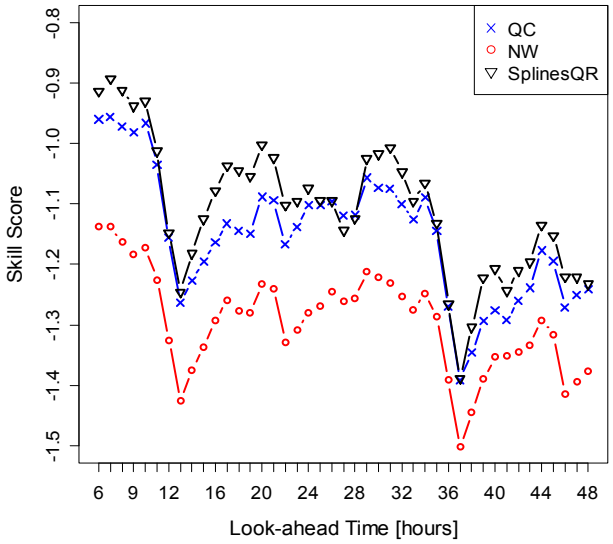

Fig. 3-295 Skill score diagram for WFB with 6:00 AM NWP and NW, QC, and QR models.

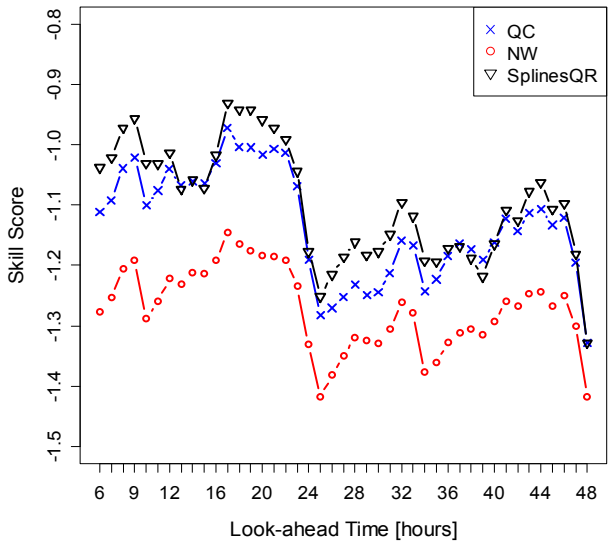

Fig. 3-296 Skill score diagram for WFB with 6:00 PM NWP and NW, QC, and QR models.

As mentioned before, the methods with better calibration present worse performance in terms of sharpness. Hence, by choosing a kernel bandwidth that increases the NW calibration performance, we were reducing the sharpness and resolution performance. If a different kernel bandwidth was considered for the variables, then the performance in terms of calibration would likely decrease; however, performance in the other metrics would increase significantly.

The following bandwidths were considered: 0.04 for wind power, 0.1 for wind speed, and 0.1 for the look-ahead time step. The results are depicted in Fig. 3-297 to Fig. 3-304. With these bandwidths, the calibration performance decreased but were still better than QC overall. The improvement was verified in both sharpness and calibration, which allowed a significant increase in the skill score. The skill score performance of KDF is competitive with the one for QR.

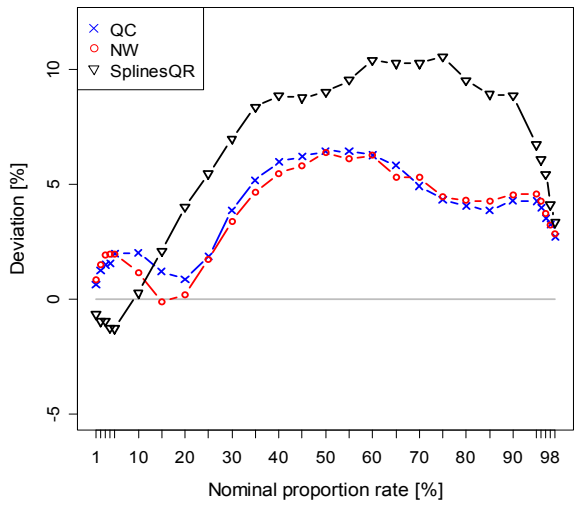

Fig. 3-297 Calibration diagram for WFB with 6:00 AM NWP and NW, QC, and QR models.

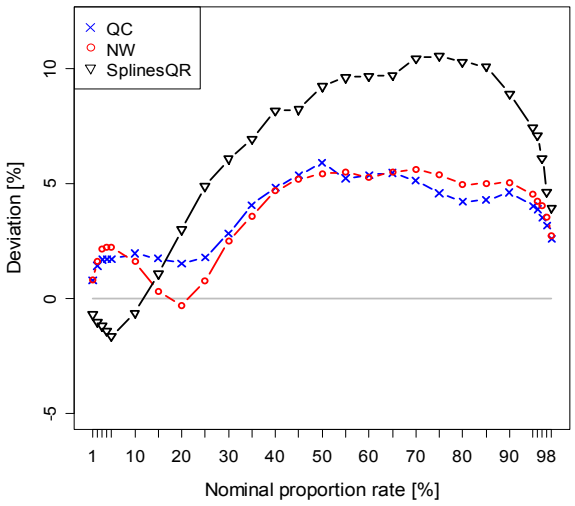

Fig. 3-298 Calibration diagram for WFB with 6:00 PM NWP and NW, QC, and QR models.

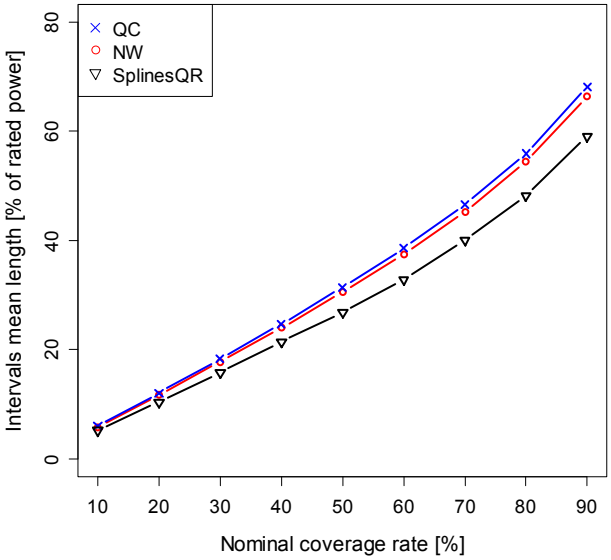

Fig. 3-299 Sharpness diagram for WFB with 6:00 AM NWP and NW, QC, and QR models.

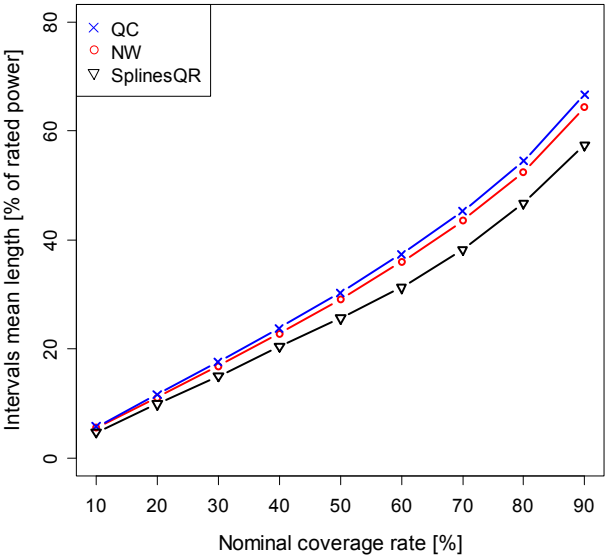

Fig. 3-300 Sharpness diagram for WFB with 6:00 PM NWP and NW, QC, and QR models.

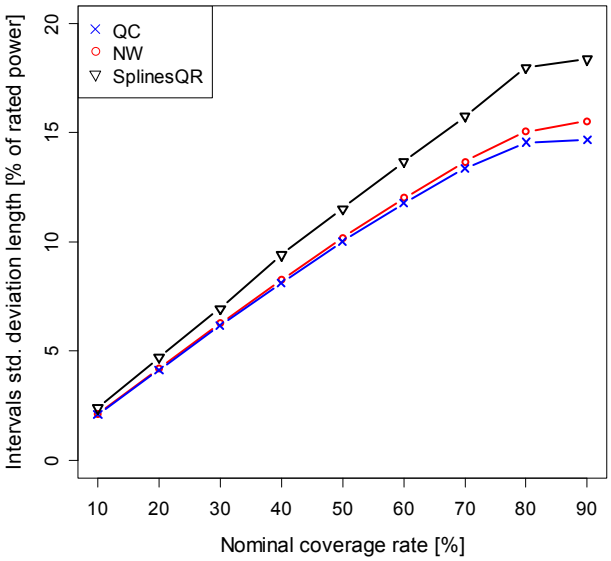

Fig. 3-301 Resolution diagram for WFB with 6:00 AM NWP and NW, QC, and QR models.

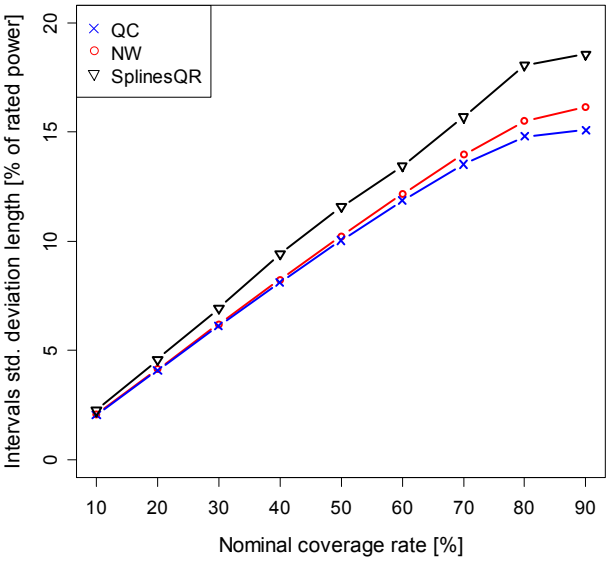

Fig. 3-302 Resolution diagram for WFB with 6:00 PM NWP and NW, QC, and QR models.

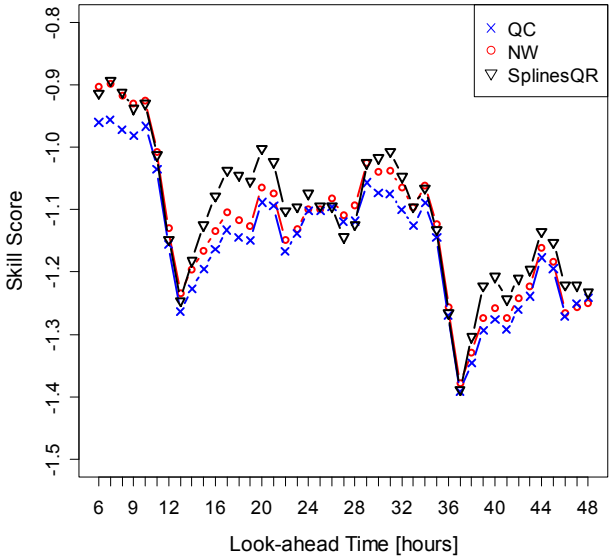

Fig. 3-303 Skill score diagram for WFB with 6:00 AM NWP and NW, QC, and QR models.

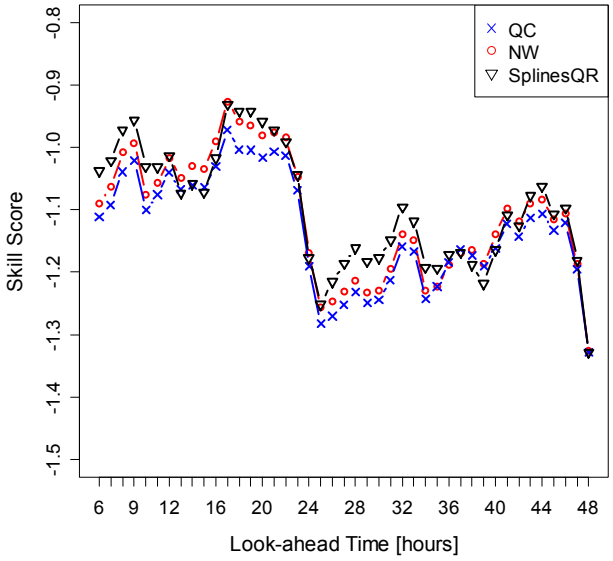

Fig. 3-304 Skill score diagram for WFB with 6:00 PM NWP and NW, QC, and QR models.

3.4.3.6 48 Hours-Ahead Time-adaptive Evaluation Results

In this section, in order to test the time-adaptive model with three inputs, the M5 model (wind speed + look-ahead time step + wind direction) was considered. The NWP launched at 6 AM was used as input.

The time-adaptive NW version was compared with the offline version for different values of the forgetting factor (λ). For a better understanding of the meaning associated with different λ

values, λ was represented by the corresponding *n* value according to (3-30). Thus, three values for λ were considered: 0.99963477 (which corresponds to n=2,738 points), 0.999 (which corresponds to n=1,000 points), and 0.995 (which corresponds to n=200 points).

Nadaraya-Watson (NW) KDF

The same kernel and bandwidths used in subsection 3.4.2.2 for the offline version was also considered for the time-adaptive versions.

Fig. 3-305 depicts the calibration results for wind farm A. The time-adaptive version with λ=0.99963477 (n=2,738 points) and λ=0.999 (n=1,000 points) achieved the best performance, whereas having a small number of points in the sliding window leads to a worse performance when comparing to the offline results. When the calibration is computed by the look-ahead time step, as Fig. 3-306 depicts for *t+20h* and Appendix D for *t+15h* and *t+10h*, the same results are verified. Results with the lowest λ present the worst performance.

The version with higher λ does not have a significant impact on the sharpness (depicted in Fig. 3-307) and resolution (depicted in Fig. 3-308). Note that the version with 200 points presents the best resolution performance.

Fig. 3-309 depicts the skill score for the offline and time-adaptive versions. The best performance was obtained with the time-adaptive versions with 2,738 and 1,000 points, whereas the version with 200 points presents the worst performance. The difference between the offline and time-adaptive versions is only noticeable in the first 28 look-ahead time steps.

Fig. 3-310 depicts the calibration results for wind farm B. For this wind farm, the best performance is for 1,000 points; however, when the calibration is computed for a look-ahead time step (*t+20h* in Fig. 3-311), the version with 200 points presents the best results; the same is valid for *t+15* depicted in Appendix E.

Fig. 3-312 and Fig. 3-313 depict the sharpness and resolution for wind farm B. Only the time-adaptive version with 200 points presents a different sharpness performance (the worst), whereas this version presents the best resolution performance.

Fig. 3-314 depicts the skill score for wind farm B. An interesting result is that the time-adaptive version with 200 points presents the best performance for several look-ahead time steps. As an example, the evaluation results for look-ahead time step *t+40h* are depicted in Fig. 3-315 to Fig. 3-317. As depicted in these figures, the version with 200 points has the best resolution and calibration performance for this look-ahead time step; consequently, this result will lead to a higher skill score.

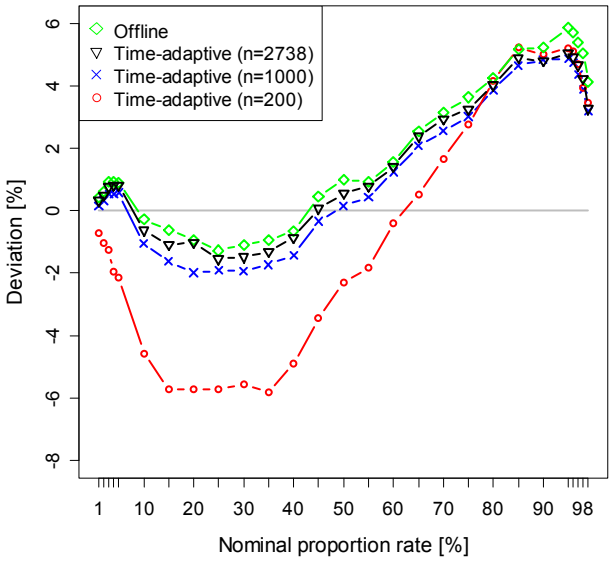

Fig. 3-305 Calibration diagram for the NW time-adaptive model with WFA dataset.

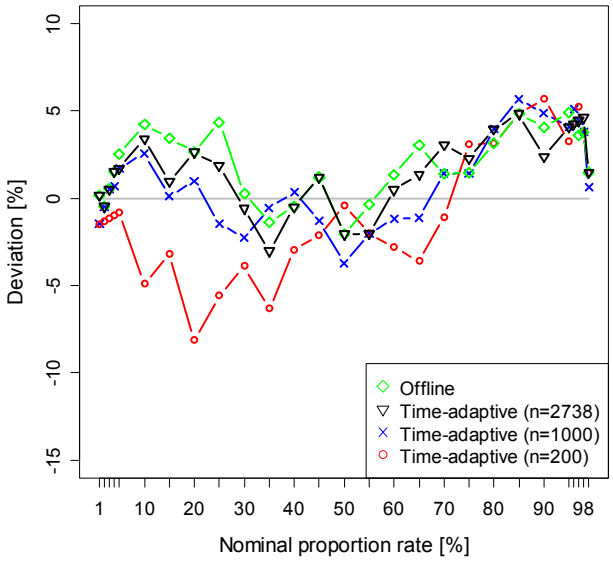

Fig. 3-306 Calibration diagram for t+20h obtained with the NW time-adaptive model for the WFA dataset.

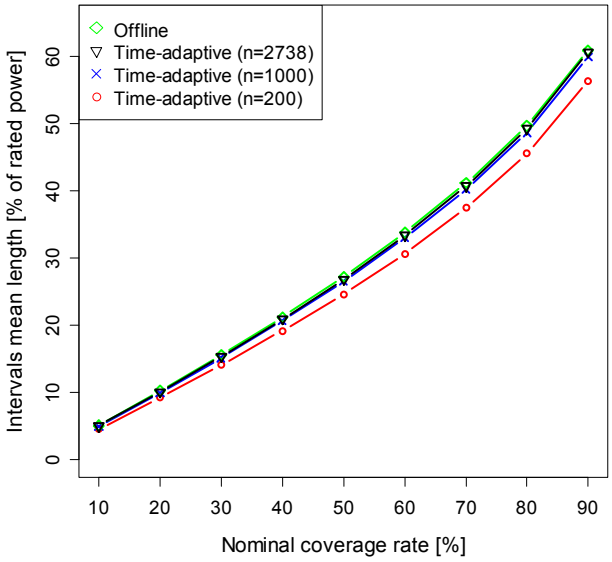

Fig. 3-307 Sharpness diagram for the NW time-adaptive model with WFA dataset.

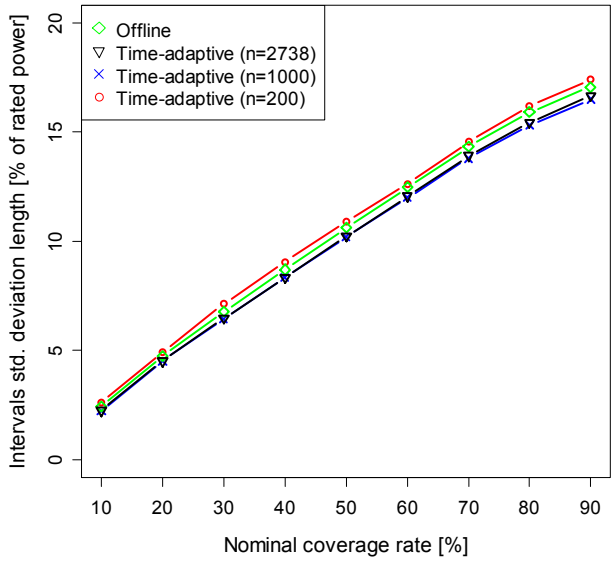

Fig. 3-308 Resolution diagram obtained with the NW time-adaptive model for the WFA dataset.

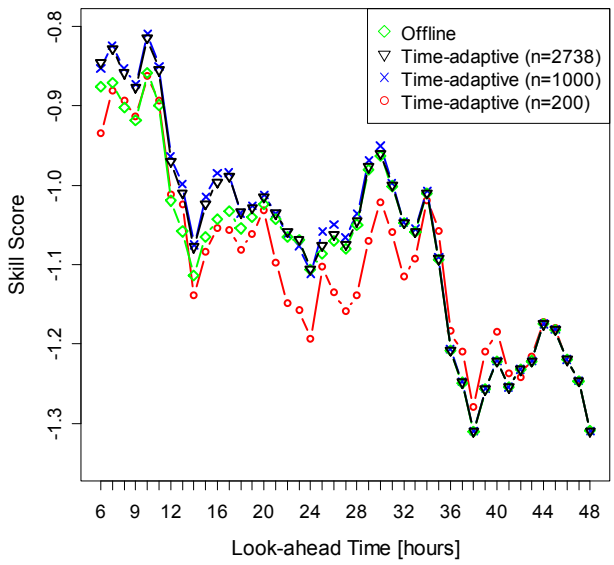

Fig. 3-309 Skill score diagram for the NW time-adaptive model for the WFA dataset.

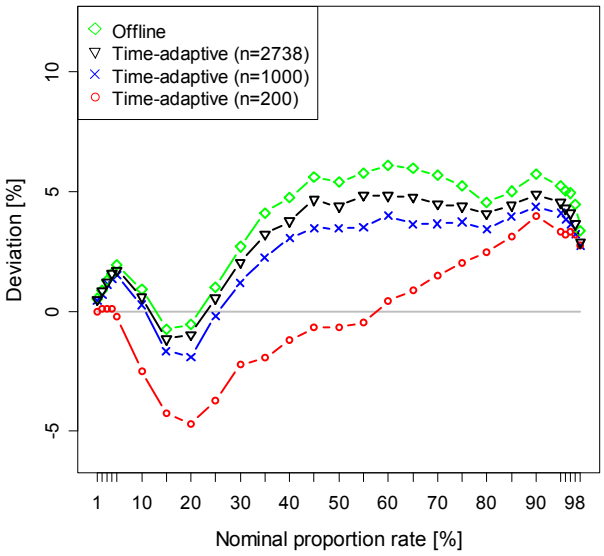

Fig. 3-310 Calibration diagram for the NW time-adaptive model with WFB dataset.

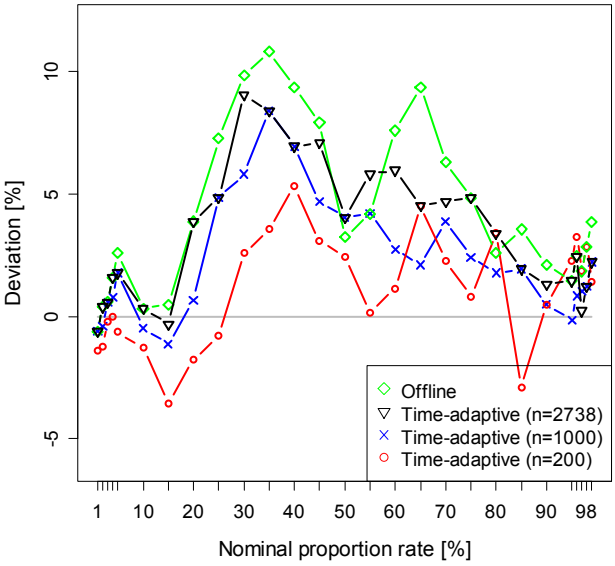

Fig. 3-311 Calibration diagram for t+20h obtained with the NW time-adaptive model for the WFB dataset.

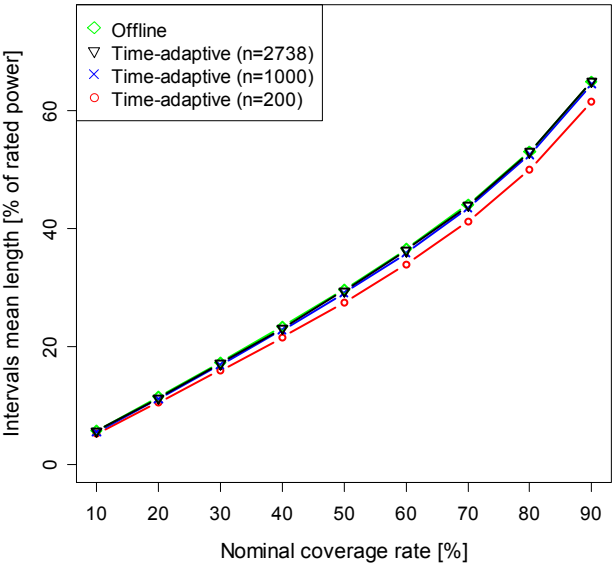

Fig. 3-312 Sharpness diagram for the NW time-adaptive model with WFB dataset.

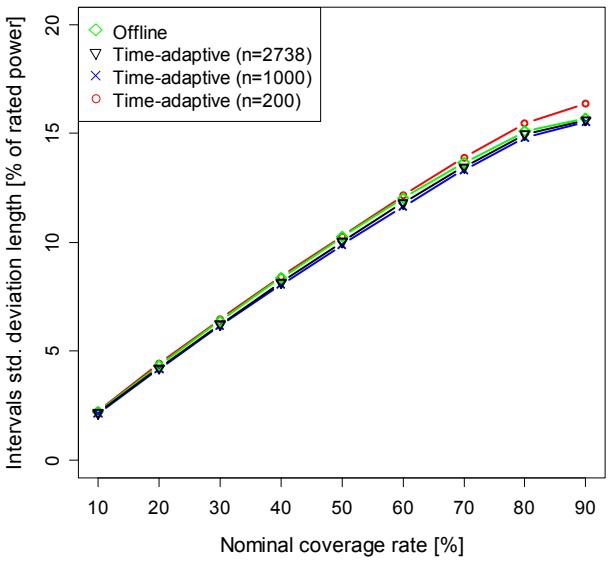

Fig. 3-313 Resolution diagram for the NW time-adaptive model with WFB dataset.

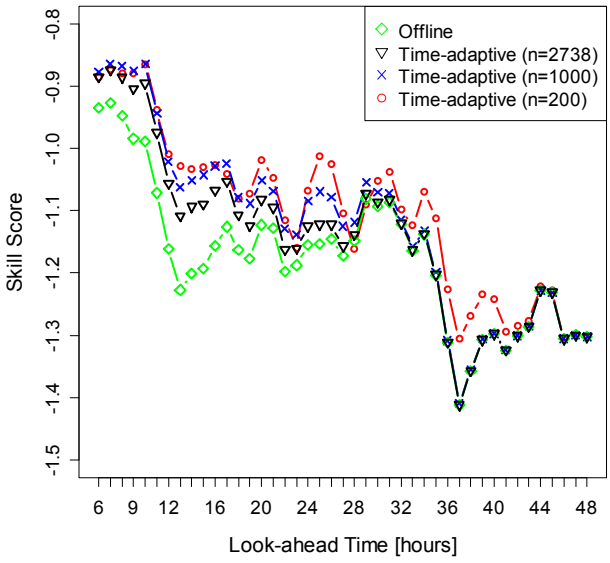

Fig. 3-314 Skill score diagram obtained with the NW time-adaptive model for the WFB dataset.

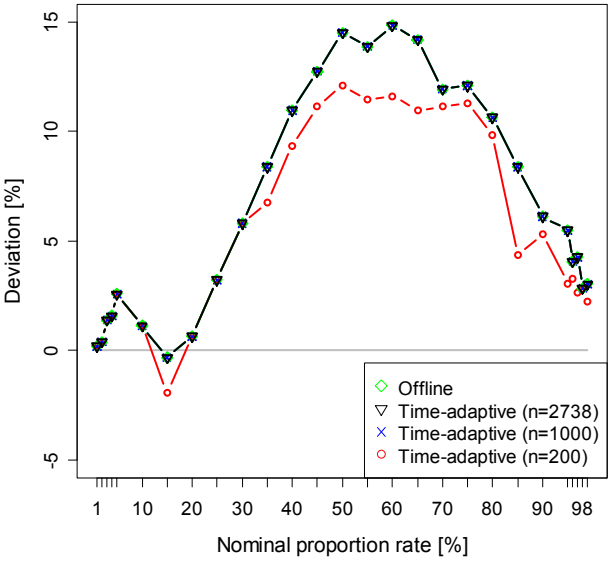

Fig. 3-315 Calibration diagram for t+40h obtained with the NW time-adaptive model for the WFB dataset.

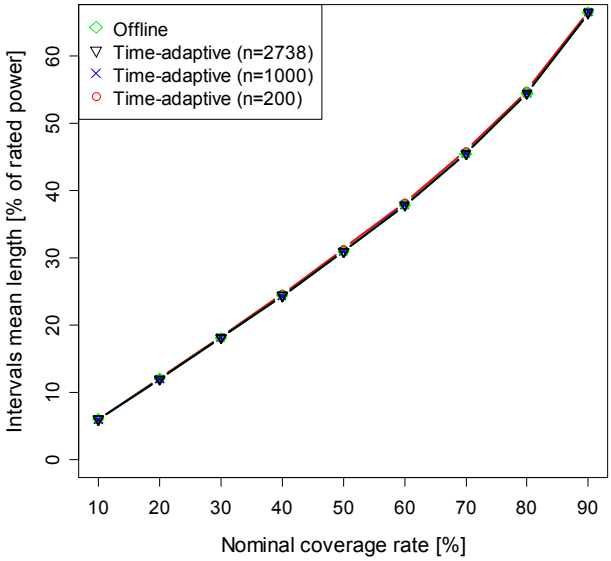

Fig. 3-316 Sharpness diagram for t+40h obtained with the NW time-adaptive model for the WFB dataset.

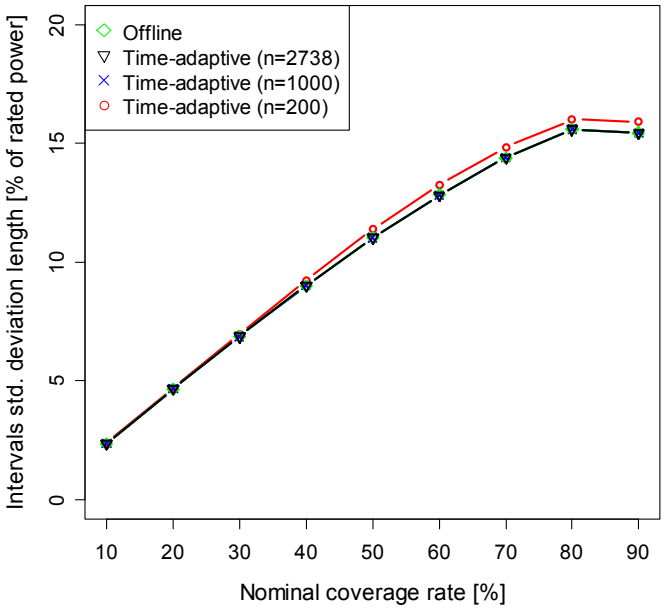

Fig. 3-317 Resolution diagram for t+40h obtained with the NW time-adaptive model for the WFB dataset.

One conclusion that can be derived from these results is that the time-adaptive approach changes calibration, or in other words, it changes the bias of the probabilistic forecasts. This change in probabilistic bias is performed in a rather uniform fashion for each quantile (it is almost a linear shift); however, for the version with 200 points, this change is not uniform.

Quantile-Copula (QC) KDF

The same kernel and bandwidths used in Section 3.4.3.5 for the offline version were also considered for the time-adaptive versions. Note that the time-adaptive version of the empirical cumulative distribution function (Eq. 3-34 in Section 0) has a different λ value. Because this dataset does not have significant variations in the data structure (in contrast to the dataset used in Section 3.4.2.3), the adopted value was 0.9999. Note that a smaller value would lead to very poor results.

Fig. 3-318 depicts the calibration results for wind farm A. The time-adaptive version with λ=0.99963477 (n=2,738 points) and λ=0.999 (n=1,000 points) achieved the best performance. The same results are obtained by the look-ahead time step; Fig. 3-319 depicts this result for *t+20h* and Appendix D for *t+15h* and *t+10h*.

The version with higher λ does not have a significant impact on sharpness (depicted in Fig. 3-320) and resolution (depicted in Fig. 3-321). Note that the offline and time-adaptive version with 200 points presents the best performance of resolution.

Fig. 3-322 depicts the skill score for the offline and time-adaptive versions. The best performance was obtained with the time-adaptive versions with 2,738 and 1,000 points, whereas the version with 200 points presents the worst performance.

Fig. 3-323 depicts the calibration results for wind farm B. For this wind farm, the best performance is for 1,000 points; however, when calibration is computed for a look-ahead time step (*t+20h* in Fig. 3-324), the version with 200 points presents the best results; the same is valid for *t+15h* depicted in Appendix E.

Fig. 3-325 and Fig. 3-326 depict the sharpness and resolution for wind farm B. The conclusions are similar to the ones derived for wind farm A.

Fig. 3-327 depicts the skill score for wind farm B. An interesting result is that the time-adaptive version with 200 points presents the best performance for several look-ahead time steps. This result occurs for the hours where the calibration performance is better, such as at the look-ahead time steps *t+20h* and *t+15h*. The worst performance is from the offline version.

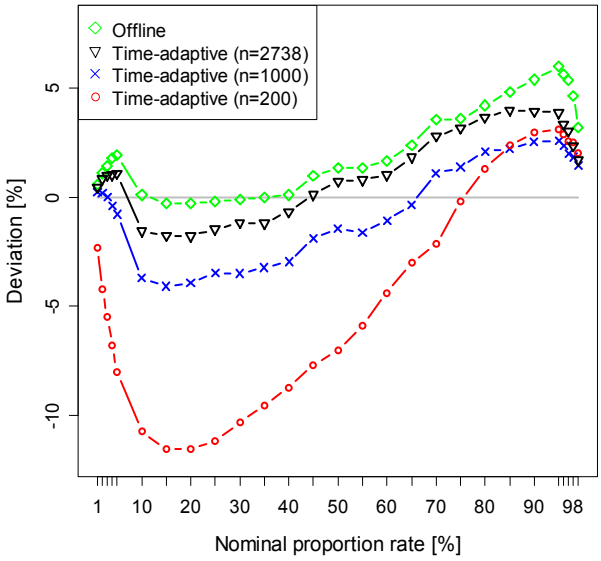

Fig. 3-318 Calibration diagram for the QC time-adaptive model with WFA dataset.

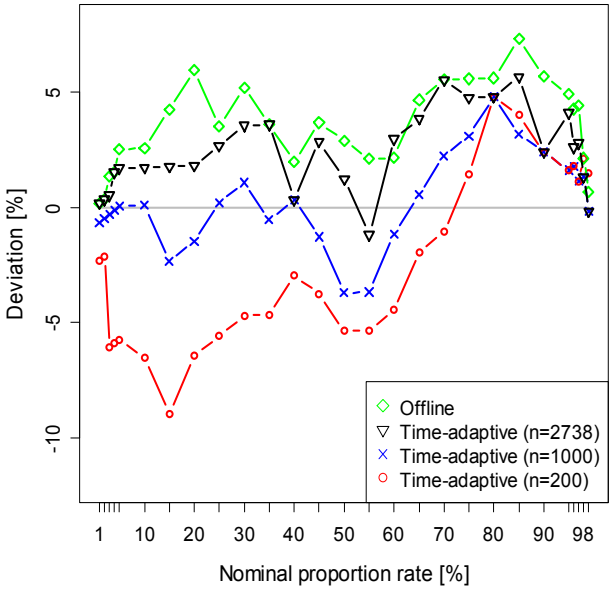

Fig. 3-319 Calibration diagram for t+20h obtained with the QC time-adaptive model for the WFA dataset.

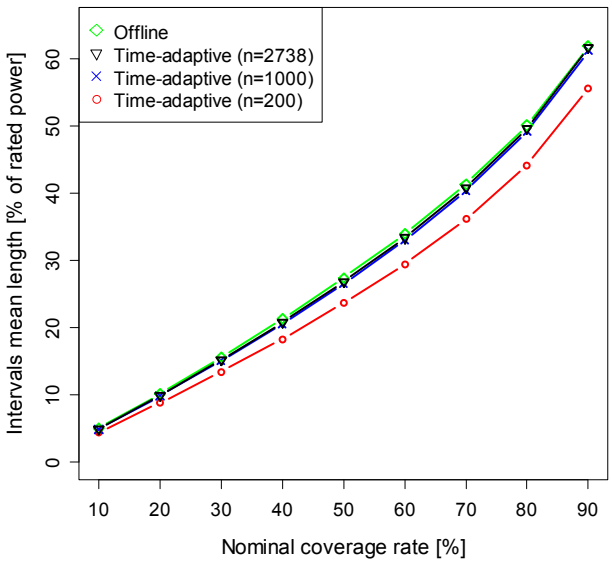

Fig. 3-320 Sharpness diagram for the QC time-adaptive model with WFA dataset.

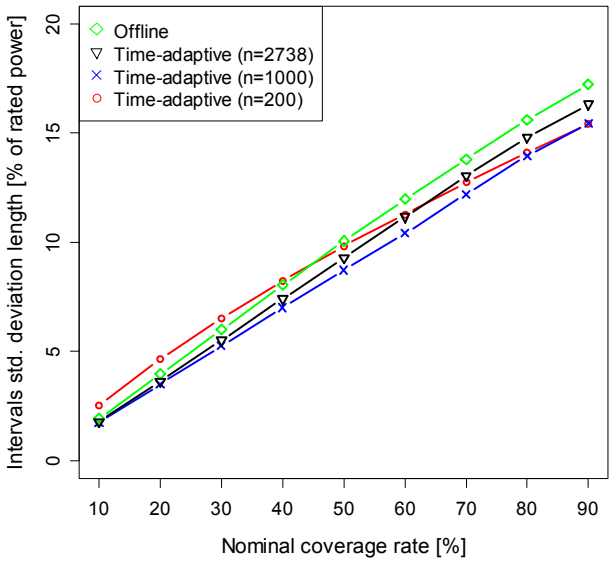

Fig. 3-321 Resolution diagram obtained with the QC time-adaptive model for the WFA dataset.

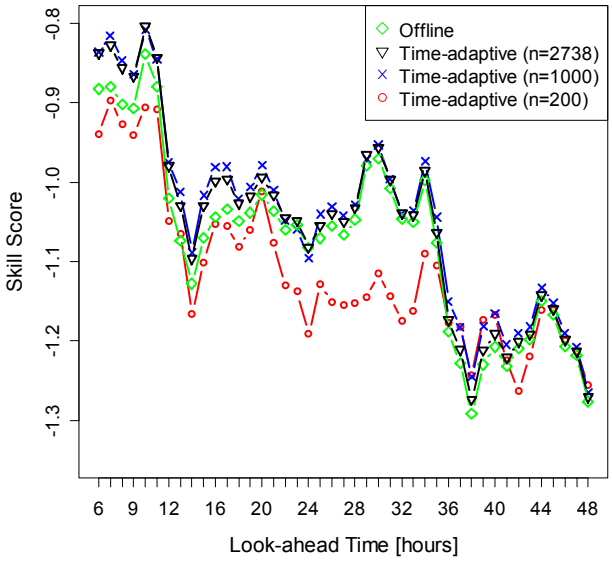

Fig. 3-322 Skill score diagram for the QC time-adaptive model for the WFA dataset.

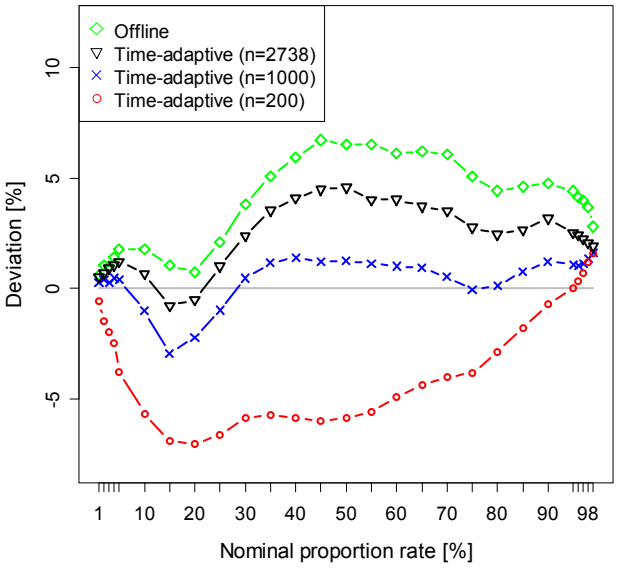

Fig. 3-323 Calibration diagram for the QC time-adaptive model with WFB dataset.

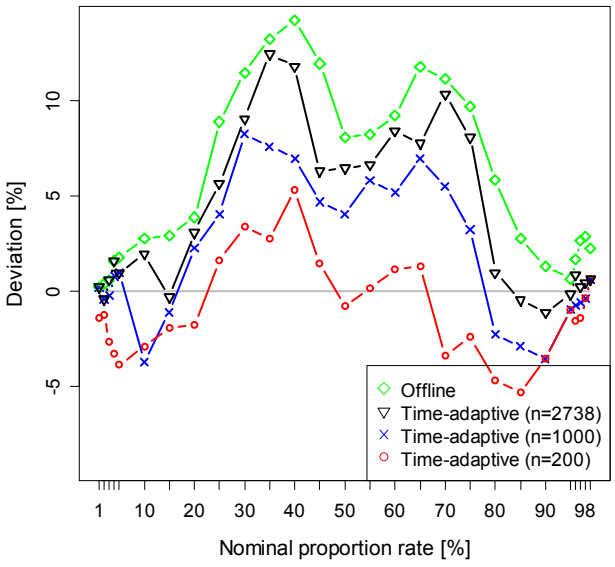

Fig. 3-324 Calibration diagram for t+20h obtained with the QC time-adaptive model for the WFB dataset.

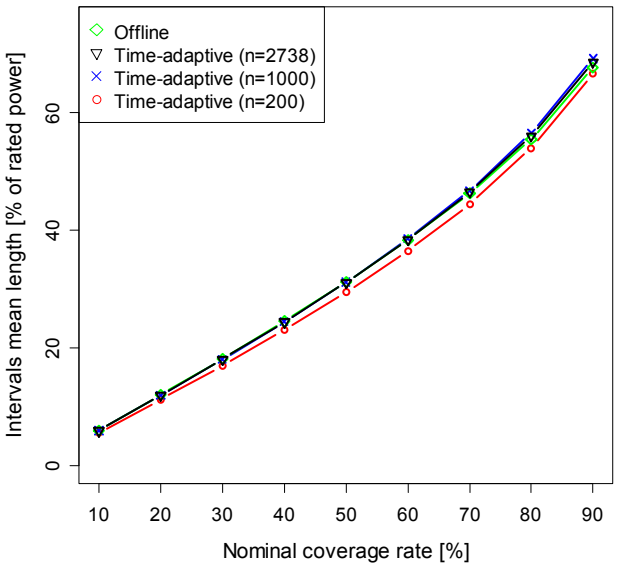

Fig. 3-325 Sharpness diagram for the QC time-adaptive model with WFB dataset.

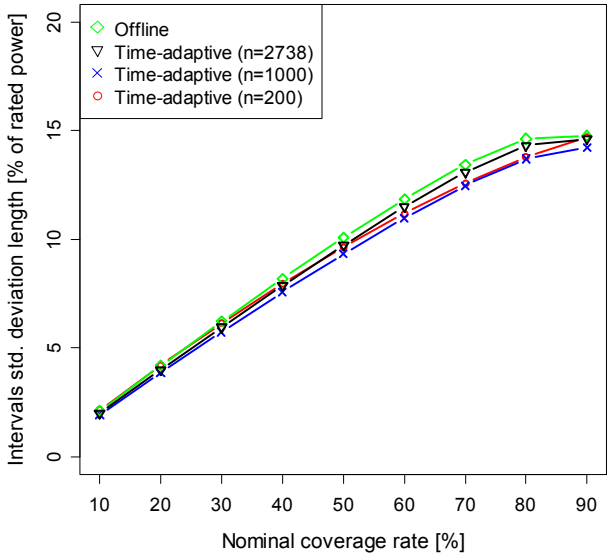

Fig. 3-326 Resolution diagram for the QC time-adaptive model with WFB dataset.

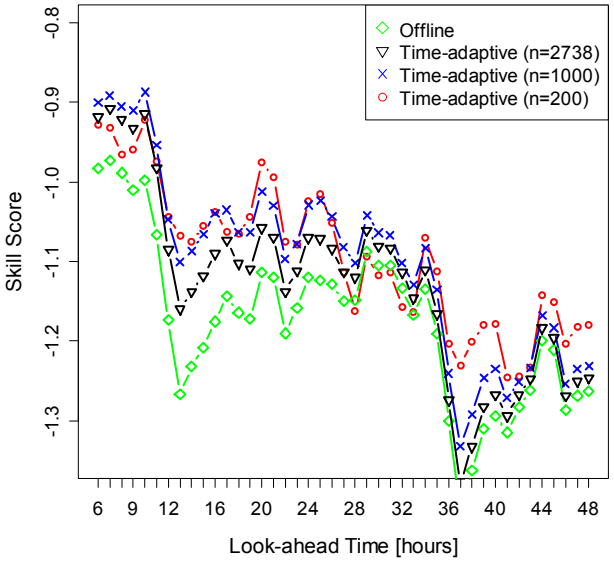

Fig. 3-327 Skill score diagram obtained with the QC time-adaptive model for the WFB dataset.

3.5 Goodness in Probabilistic Forecasts: A Discussion

Only the *quality* of the probabilistic forecast was evaluated in this chapter (Section 3.4) by using four different metrics (calibration, sharpness, resolution, and skill score). These four evaluation metrics only evaluate the correspondence between the forecasts and the reality, and no considerations are made about the impact on decision-making problems.

It is essential to compare the results also from the end-user's perspective, which consists of the additional *value* (e.g., economic, technical, psychological) introduced by the probabilistic forecasts in a particular decision-making problem. In the WPF literature, there are three main decision-making problems: (i) finding the "optimal" wind power bid for the electricity market; (ii) setting the operating reserve in systems with high penetration of wind power; and (iii) establishing unit commitment. These decision-making problems have different end-users, mainly wind power generation companies (WGENCOs) and system operators (SOs).

The evaluation of the probabilistic forecast *value* can be performed solely by the use of probabilistic forecasts in a specific decision-making problem and subsequent evaluation of the decision's *quality*. However, the *quality* metrics used in this chapter can provide hints about the forecast *value*, but the interpretation will differ from problem to problem.

In what concerns the wind power bidding problem, the most important information comes from the calibration metric, in particular for the methodologies that try to find the "optimal quantile" (for more details, refer to [20] and [59]). Under certain assumptions, in these methodologies the bid is equal to a quantile value that is found to be the "optimal" (e.g., under the expected value paradigm). Hence, the bias in the probabilistic forecast (represented by the calibration) is the motive for making "bad" and "good" bids. The same is valid for decision rules where a trade-off between expected value and risk is evaluated. For instance, if the risk is represented by the value at risk (which is a quantile of the *pdf*), a huge probabilistic bias may lead to an under- or overestimation of the risk; moreover, only a good calibration guarantees the accuracy in the expected value computation. The sharpness and resolution are not very relevant for this problem.

For the bidding problem the probabilistic forecasts produced with KDF methods present better calibration and are therefore likely to have a higher *value* compared to the ones obtained with QR methods.

The *value* of probabilistic forecasts used as input for setting an operating reserve (for more details, refer to [24] and [60]) is contained in the three metrics. The SO in this problem normally sets a reference value for a risk index; for example, ERCOT (the Electric Reliability Council of Texas, i.e., the Texas Independent System Operator) defines a nonspinning reserve corresponding to quantile 95% of the historical total forecast error (i.e., the same as setting a loss of load probability equal to 0.05). Hence, the bias of the probabilistic forecasts (calibration) provides information about the under- and overestimation of the risk [25]. Probabilistic forecasts with considerable bias value may lead to decisions with very bad consequences (e.g., power system blackouts) and represent a source of stress to the operators. As an example, the operator may choose an operating reserve equal to the 1% quantile of the system generation margin distribution (distribution of the difference between generation and load), but because of the bias in the wind power uncertainty forecast, this quantile may actually be 5%, meaning that the real probability of loss of load is 0.05 (instead of just 0.01).

According to Matos and Bessa [24], the shape of the forecast distribution has an impact on the operating reserve requirements. Therefore, sharpness and resolution that are measures for the shape of forecast distributions are also important factors. Moreover, a forecast with higher sharpness means a forecast with a higher "amount of uncertainty." In the operating reserve

problem, this distribution leads to uncertainty intervals with higher amplitude, which could represent an increase in the reserve values.

However, this behavior is not yet studied; Bessa and Matos [25] only studied the calibration property. It is also important to stress that forecasts with a higher value for sharpness can cover extreme events (improbable, with very bad consequences).

For the operating reserve problem, apparently the calibration is the most important factor, so the probabilistic forecasts produced with KDF methods could have a higher *value* as compared to those obtained with QR methods. However, we should emphasize again that the sharpness and resolution in probabilistic forecasts may also be an important factor in this problem.

Unit commitment is a time-dependent decision-making problem, so that the wind power uncertainty should be represented by wind power scenarios that respect the probabilistic forecasts [26]. The *quality* of decisions is related with scenario *quality*, and the quality of the scenarios is related with the probabilistic forecast *quality*. In general, probabilistic forecasts with higher calibration and sharpness would lead to better representation in terms of scenarios. As mentioned by Pinson *et al.* [26] regarding the probabilistic forecast quality: "If these marginal distributions used as input were not reliable, the generated scenarios would also not be reliable." Hence, the aim is to have probabilistic forecasts that are a good compromise between calibration and sharpness. The results presented in this chapter show that KDF methods provide a satisfactory compromise between these two metrics.

Finally, the KDF methods have the possibility of controlling the calibration by changing the kernel's size. This flexibility could be considered an advantage over the QR methods for some problems (the bidding problem, for instance).

3.6 Conclusions

The KDF models present results consistent with what is found in the wind power uncertainty forecast literature.

From the results and sensitivity analyses performed in Sections 3.4.3.2, 3.4.3.3, and 3.4.3.4, it is possible to derive the following conclusions:

- Chen's gamma and beta kernels from (3-17) and (3-21) have better performance not only in calibration but also in skill score;
- Different dataset characteristics lead to distinct results in calibration and skill score;
- Different wind speed kernel sizes (either lower or larger than 1) lead to distinct results in terms of the resolution and skill score;
- The KDF methods have a tendency to present a better performance in terms of calibration in WFB;
- All of the kernels presented similar sharpness and resolution results, although QR methods tend to have a better performance; and
- The skill scores of splines QR and NW and of linear QR and QC are rather similar.

From the complete case study presented in Sections 3.4.3.5 and 3.4.3.6, it is possible to derive the following conclusions:

- The KDF methods have a tendency to present better performance in terms of calibration;
- The QR methods have a tendency to present better performance in terms of sharpness and resolution;
- The skill scores of the QR and KDF methods are rather similar;
- The time-adaptive approaches for both NW and QC changes the bias of the probabilistic forecasts (calibration), while changing slightly the sharpness and resolution; and
- The time-adaptive approach improves the skill score when compared with the offline approach.

The main contributions to the state-of-the-art from the two models described in this chapter are the following:

- In comparison to the KDF model developed by Juban *et al.* [17], the NW estimator described in this chapter is time-adaptive;
- The model presented by Juban *et al.* [17] is an adaption of the classic NW estimator, whereas our approach is different and based on selecting the adequate kernel for modeling the different variables in the wind power forecast problem. Our proposal is simple and provides enough robustness to deal with any type of variables, such as using circular kernels (e.g., von Mises distribution) for circular variables (e.g., wind direction); and
- The QC approach was applied for the first time to the wind power forecasting problem and also with circular variables. Moreover, a time-adaptive version of the method is described and evaluated.

As future developments, we envision the following research topics:

- Inclusion of wind power measurements in the uncertainty forecast model in order to improve very short-term forecasts (e.g., forecasts for look-ahead time steps below 6 hours);
- Development of an adaptive strategy for the forgetting factor (λ);
- Development of heuristic rules for setting the kernel bandwidths;
- Development of an evaluation scheme that accounts for the end-users' preferences and ideas of "good forecasts," for example, a skill score oriented for a decision-making problem.

This page intentionally blank

4 WIND POWER RAMP FOREACSTING: A PROPOSAL

This chapter presents a new method for ramp event detection and a comparative study about the existing definitions and methodologies to forecast ramp events. The paper is structured as follows. After an introductory section, Section 4.2 describes the formal and informal terms defined in the literature. Section 4.3 presents a new methodology, based on scenarios, for probabilistic ramp event detection and visualization. Section 4.4 illustrates a comparative study of the new proposal using the terms defined in the literature. Section 4.5 summarizes the conclusions stated through the study.

4.1 Introduction

One of the major issues in wind power generation are ramp events. These events are characterized by sudden and wide changes, either an increase or a decrease, of wind power generation. In the presence of such events, system operators (SOs) have to develop operational procedures in order to satisfy the load and maximize both the economical and environmental benefits. The longer the time-ahead prediction of such events, the higher the uncertainty and effectiveness of such procedures. To deal with a ramp-up event, a wind power producer may have to reduce generation, according to its market commitments; or the SO may use downward spinning reserve to compensate these ramps. During a ramp-down event, the SO will typically need to activate fast upward spinning reserve (i.e., switch on fast start-up units) [61], increasing the system operating cost.

Recently, with the dissemination of modern turbine technology and large wind farms, the percentage of energy from wind sources relative to the peak load is rapidly increasing in many parts of the world. Thus, the demand for more reliable wind power is pushing up the critical need for ramp events detection and prediction [62]. For instance, [63] reported a rapid and large ramp-down event in the Electric Realiability Council of Texas (ERCOT) area on February 26, 2008, that forced ERCOT to declare system emergency, with a high-cost system condition. This type of event underscores how crucial having an accurate forecast of ramp events and quantification of ramp forecast accuracy is to the large-scale integration of wind energy into electrical grids, and also helps market participants better understand the risk involved in trades at times of high variability [64].

One of the main problems in ramp forecasting is how to define a ramp. In fact, there is no standard formal definition of it [62][65][66], and almost all existing literature reports different definitions, depending, for instance, on the location or on the farm's size.

4.2 Ramp Event Definitions

Ramp forecasting is a relatively new research field. In order to study the ramp phenomena, it is important to define what is or could be considered to be a ramp event. According to AWS Truewind's technical report for the Alberta Pilot Project [67], a ramp is a change in the power output with a large enough amplitude and over a relatively short period of time. These events may cause grid management problems in the next few hours or days. The same idea also appears in [64][68]. The expressions swings, extreme events, or rapid changes are also used and

considered as being synonymous with a ramp event [69]. Fig. 4-1 illustrates the ramp definition presented in [64]: a change in the wind farm power output of at least 50% of the installed capacity in a time span of four hours or less.

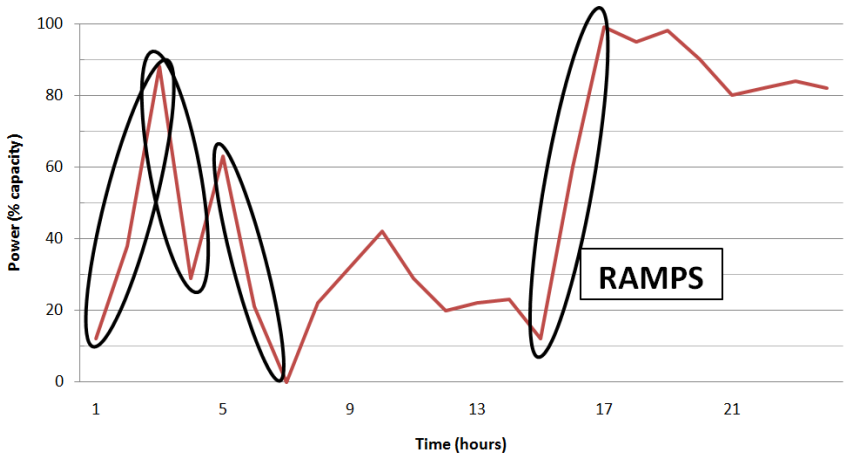

Fig. 4-1 Ramp event definition: a change in power of at least 50% of the capacity, over a maximum duration period of 4 hours (freely based on figure from [70]) .

4.2.1 Characteristics of Ramp Definitions

The authors in [64] and [71] define several relevant characteristics for ramp definition, characterization, and identification: to define a ramp, we have to determine values for its three key characteristics — direction, duration, and magnitude.

With respect to direction, there are two basic types of ramps: upward ramps (or ramp-ups), and downward ramps (or ramp-downs). The former, characterized by an increase of wind power, result from a rapid rise in wind speeds, which might (although not necessarily) be due to low-pressure systems, low-level jets, thunderstorms, wind gusts, or other similar weather phenomena. Downward ramps are due to a decrease in wind power, which may occur because of a sudden depletion of the pressure gradient, or may be due to very high wind speeds that lead wind turbines to reach cut-out limits (typically 22–25 m/s) and shut down in order to prevent the wind turbine from experiencing damage [72]. In order to consider a ramp event, the minimum duration is assumed to be 1 hour in [71], although in [62], these events lie in intervals of 5 to 60 minutes. The magnitude of a ramp is typically represented by the percentage of the wind farm's nominal power — its nameplate capacity.

Duration and magnitude are usually related. For example, [71] considers rapid ramp events when the hourly change in power output is greater than or equal to 10% of the nominal capacity of the wind farm. In addition to this definition, the AWS report [67] suggests that:

- An important downward ramp occurs only if the power change in one hour is, at least, 15% of the total capacity;
- An important upward ramp occurs if the power change in one hour is, at least, 20% of total capacity.

In the next subsection, we present some ramp event definitions using these characteristics.

4.2.2 Ramp Definitions

Although it is easy to identify ramps visually, there is no agreed-upon and/or accepted formal definition of a ramp event [62][65][66]. In this section, we present four ramp event definitions. As mentioned above, a ramp event can be characterized according to three features: direction, magnitude, and duration. However, if we consider that ramp magnitude values range from positive to negative, then we can characterize a ramp using only magnitude and duration features. The sign of the magnitude value can give us the ramp direction: positive magnitude values correspond to upward ramps, and negative magnitude values correspond to downward ramps.

In the definitions below, a ramp event can be identified according to the power signal, $\mathrm{P(t)}$, and two user-defined parameters (one of the definitions requires only one parameter to identify a ramp). The parameter Δ_t is related to the ramp duration (given in minutes or hours) and defines the size of the time interval considered to identify a ramp. In [71][73], some results are presented that relate this parameter to the type and magnitude of identified ramps. The other parameter, P_{ref}, is related to the ramp magnitude feature and provides a cut-off level on the power changes. The P_{ref} parameter is usually defined according to the specific features of the wind farm site. This threshold value depends on the amount of wind power installed, and is defined as a percentage of the nominal wind power capacity or a specified amount of MW (megawatts). In [62], the authors claim that defining the P_{ref} value according to the wind farm nominal capacity can produce unreliable results. They analyze historical measurements, considering that the nominal capacity of a wind farm is always changing: at each moment, one or several units can be turned off. They studied the sensitivity of two ramp definitions to each of the two parameters introduced above: P_{ref} ranging from 150 to 600 MW and Δ_t values varying between 5 and 60 minutes.

The first defintion that we present here has been formally described in [62].

Definition 1: A ramp event is considered to occur at the start of an interval if the magnitude of the increase or decrease in the power signal, at time Δ_t ahead of the interval, is greater than the threshold value, P_{ref}:

$$|P(t+\Delta_t) - P(t)| > P_{ref} \quad (4\text{-}1)$$

This inequality only considers the values at the end points of the interval, ignoring the ramps that occur in the middle. To address this issue, [62] extended the previous definition.

Definition 2: A ramp is considered to occur in a time interval, Δ_t, if the difference between the maximum and the minimum power output measured in that interval is greater than the threshold value, P_{ref}:

$$\max(\mathrm{P}[\mathrm{t}, \mathrm{t}+\Delta_\mathrm{t}]) - \min(\mathrm{P}[\mathrm{t}, \mathrm{t}+\Delta_\mathrm{t}]) > \mathrm{P_{ref}} \quad (4\text{-}2)$$

This inequality considers the total magnitude of the power fluctuation through the interval. However, this definition does not consider the curve's slope: that is, how fast the power output

decreases or increases. In order to analyze this important factor, we cannot consider an absolute threshold like P_{ref}, as we need a time-relative threshold.

A more elaborate definition considers the rate of change in power output over a period of time [66]. The authors define *Power Ramp Rate*, or slope, to be the rate of change of the power with respect to time. This measure is expressed in MWm^{-1} (megawatts per minute).

Definition 3: A ramp is considered to have occurred if the difference between the power measured at the initial and final points of a time interval, Δ_t, is greater than a predefined reference value to the Power Ramp Rate, PRR_{ref}:

$$\frac{|P(t+\Delta_t)-P(t)|}{\Delta_t} > PRR_{\text{ref}} \quad (4\text{-}3)$$

In the definitions presented in equations (4-1) and (4-3), we can easily identify the type of ramp: if $P(t) > P(t+\Delta_t)$, we are analyzing a downward ramp; otherwise it is an upward one. On the other hand, this distinction is not that clear in (4-2). In this latter case, we can identify the type of the ramp by using the relative position of the extreme time points within the interval. If the maximum power output occurs after the minimum power output, we have an upward ramp; otherwise, we are experiencing a downward ramp.

While the definitions above work directly with the wind power signal, other approaches transform the signal into a more appropriate representation. A usual transformation consists of considering k-order differences in the power amplitude. This strategy is used, for example, in [74]. Let p_t be the wind power time series and p_t^f the associated transformed signal that was obtained according to

$$p_t^f = mean\{p_{t+h} - p_{t+h-n_{am}}; h = 1, \dots, n_{am}\} \quad (4\text{-}4)$$

In this formula, the parameter n_{am} stands for the number of averaged power differences to be considered.

Definition 4: A ramp event is said to occur in an interval if the absolute value of the filtered signal, p_t^f, exceeds a given threshold value, P_{ref}:

$$|p_t^f| > P_{\text{ref}} \quad (4\text{-}5)$$

If required, the ramp time is considered to be the interval point for which the filtered signal has its maximum magnitude. This definition was introduced in [74].

Definition 5: This definition is based on filter techniques from signal processing to remove unwanted frequency components from a signal. We have developed this new definition under this project. It uses a high-pass filter, that is, a filter that passes high-frequency signals and attenuates (reduces the amplitude of) signals with frequencies lower than the cut-off frequency. The simpler high-pass filter can be formulated as:

$$y[i] = \alpha(y[i-1] + x[i] - x[i-1]) \quad (4\text{-}6)$$

It can only pass relatively high frequencies because it requires large (i.e., fast) changes and tends to forget its prior output values quickly (see Fig. 4-2) . A large α implies that the output will decay very slowly but will also be strongly influenced by even smaller changes in the input signal. A constant input (i.e., an input with $x[i] - x[i-1] = 0$) will always decay to zero. A small α implies that the output will decay quickly, requiring large changes in the input (i.e., $x[i] - x[i-1]$ is large) so that the output varies considerably.

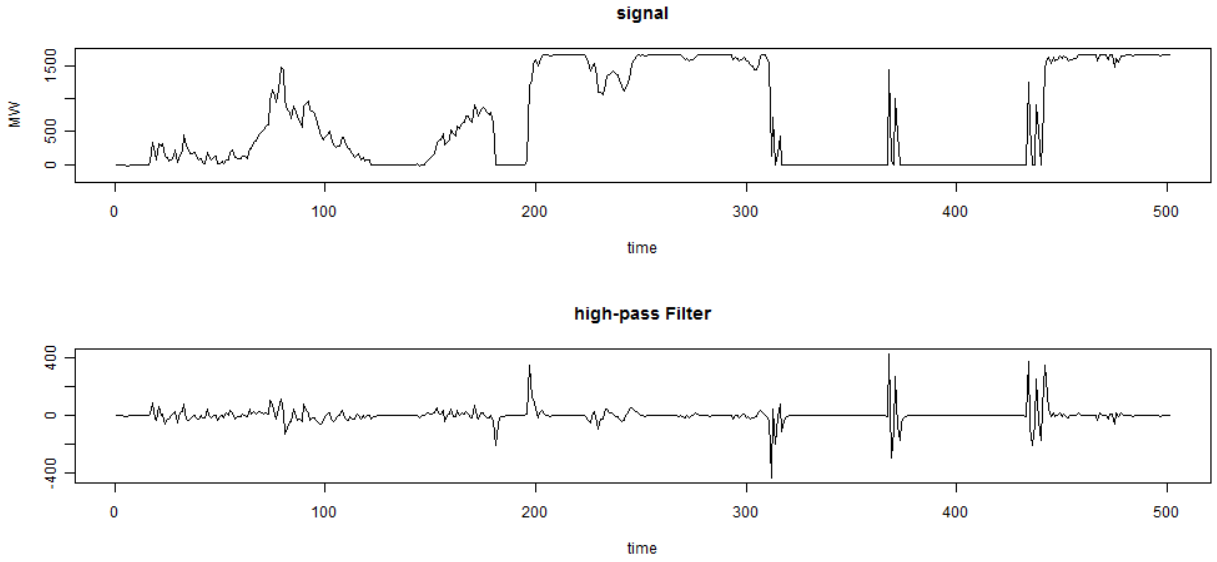

Fig. 4-2 Example of applying Definition 5, using a high-pass filter. The top panel presents the original signal. The bottom panel represents the high-pass filtered output signal. The output signal increases only when there is a fast variation of the input signal.

Small variations in the output signal might be removed using a band filter. A band filter sets to 0 all data points whose absolute value is smaller than an admissible threshold. Filter-based ramp detection can be applied using the predictions for the total park, for individual turbines, or forecast scenarios as the input signals. The filtered signal from different scenarios (or turbines) are aggregated using histograms that will be explained in the following section.

4.3 New Methodology for Detecting Ramp Event Probability

4.3.1 Basic Ideas for a New Model

The construction of a model for ramp event prediction has a precise guideline. The goal is to produce a model able to supply decision models with data of a probabilistic nature, such that generator scheduling and dispatch decisions may be made in a framework of risk analysis — taking into account the trade-off between risk of loss because of unexpected ramping that the system cannot cope with and the cost of hedging against such risk.

Although there are no good industrial models that would directly produce such a risk-minded generator schedule, this approach is seen as the most promising line of development in decision models for system operation, given a high penetration of volatile or intermittent resources.

Therefore, the objective of the new model is to produce as results, for each hour of the generator scheduling/unit commitment exercise, information regarding the possibility of occurrence of a ramp event in the form of probabilities up or down.

The simplest form is to have a single definition of a ramp (taken from some industry definition), either based on the rate of change in wind power and the duration of change, or based on a certain magnitude of change within a given time interval — producing, for each hour, a probability of occurrence of such an event.

A more complex output may combine a series of ramp definitions into a ramp probabilistic distribution for each hour. Hence, this distribution represents the probability of having a ramp event of a certain magnitude or higher for a range of magnitudes.

The departing point to build such a model will be a probabilistic description of the wind power forecast. In this chapter, we adopt a wind power scenario generator as used in [75], which can be considered the state-of-the-art and that has been used in other tasks of the project, such as in a model to supply wind power scenarios for unit commitment.

Although this chapter does not discuss the quality of this scenario generator model, it should be kept in mind that its quality conditions significantly affect the quality of the ramp forecasting exercise. The scenarios used in this study have been generated according to the methodology introduced by Pinson et al. [76], which is equivalent to producing scenarios under a Monte Carlo process.

The new model is not intended to produce directly the input to a unit commitment program. Regarding the concept developed at INESC Porto together with Argonne, this input is under the form of scenarios entering as data in a stochastic programming application; and this input will generate a schedule that, in the probabilistic sense, will define the optimal decisions.

The set of scenarios is condensed from a discrete representation of the probability density function (pdf) associated with the wind power forecast. This pdf already has information on the frequency of occurrence of ramps at any hour — therefore, an optimization of the unit commitment with a stochastic programming model will take into account this factor. So, why do we specifically need a mechanism to forecast ramp events?

This (new) model serves two purposes. First, it can be used as a tool to generate alarms and allow the system operator to pay particular attention to specific hours. Complementing this purpose, the system operator has more information in order to allow hedging against dangerous ramp events — by adjusting the schedule to meet some target of risk that can now be quantified.

Second, it may be used to check and validate the generator schedules proposed by the unit commitment algorithm. One must bear in mind that the number of scenarios supplied as input of

a stochastic optimization model must necessarily be small, for reasons of computing feasibility. There is a gross discretization of the continuous representation of the pdf, which implies unavoidable approximations. A tool that allows the verification of the proposals from a complex optimization process seems valuable. Otherwise, some unit commitment outcomes might be regarded as obscure or incomprehensible to system operators, because these outcomes are derived from a compromise within a set of scenarios; or as being difficult in terms of perceiving its full implications, in contrast to the straightforwardness of a deterministic strategy based on a point forecast.

In a nutshell, the scenario generator model is able to produce likely scenarios for the evolution of the wind power in the coming 24 or 48 hours. These scenarios must be seen as fair drawings of a Monte Carlo process. By examining the scenarios and counting under a comparison with a target ramp shape (or definition), one may calculate the sample probability of encountering a ramp of a given shape or magnitude at each hour. This calculated probability also allows one to build cumulative probability functions, describing for each threshold, P, the probability of having a ramp event equal or greater than P at a given hour or defined time step.

The integration of the new model in the wind power forecasting landscape now becomes easy to understand, and Fig. 4-3 illustrates this integration: an implicit model for the pdf of the wind power is a common model, both for ramp event analysis and for unit commitment. This model requires a reduced set of scenarios that must be generated as representatives of a much larger set. The large set is used directly to produce a probabilistic description of ramp events, which may be coupled with the output of the unit commitment module in order to allow validation and possible hedging.

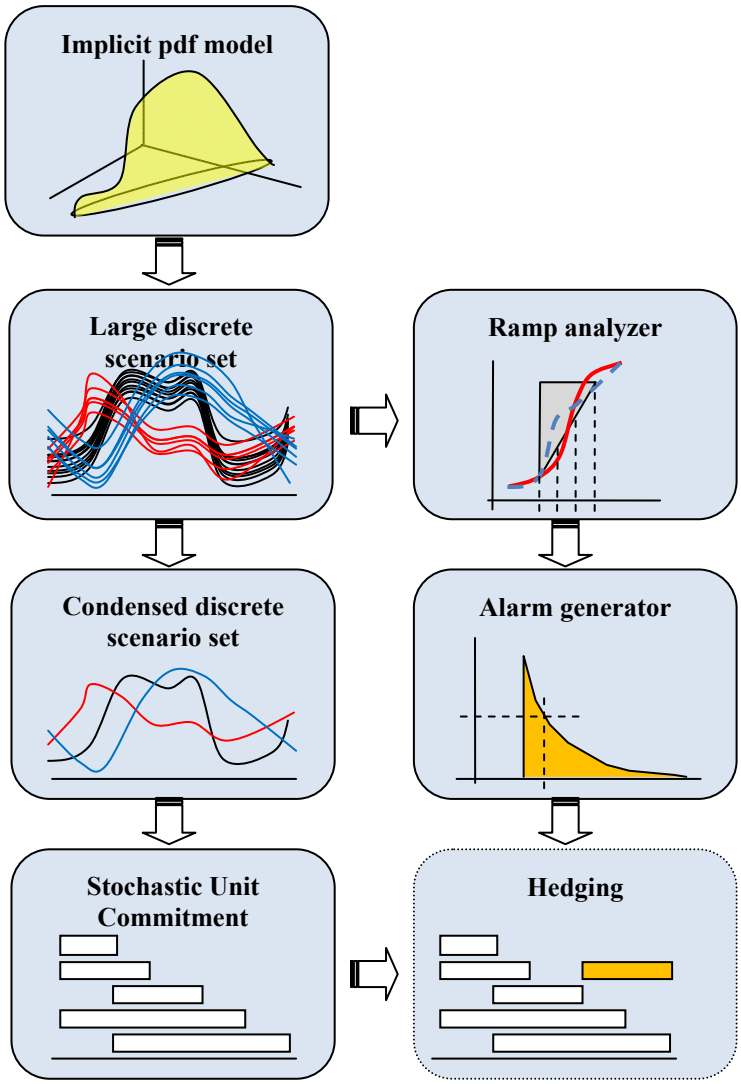

Fig. 4-3 Conceptual modules relating scenario generation, unit commitment, and ramp event analysis.

4.3.2 Development

This chapter opens the discussion and proposes a new way to detect and represent the possibility of ramping events in short-term wind power forecasting. Ramping is a remarkable characteristic in a time series associated with a drastic value change in a set of consecutive time steps. In the context of system operation, three properties are important to define the perception of the event: the amplitude of change, the duration, and the probability of occurrence at a given time step. Amplitude and duration are associated with the slope. The probability of occurrence is associated with phase error in prediction. Ramping may be classified as up or down, depending on the sign of the slope. Phase errors may be of advancement or delay, relative to a specific point forecast.

Both properties (slope and phase error) are important from the point of view of one of the main users of wind power forecasts: the SO. Thus, these properties may have important implications in

the decisions associated with unit commitment or generation scheduling, especially if there is thermal generation dominance in the power system. Most large thermal generators cannot start up under short notice, nor can they change their output at a fast pace (otherwise, a severe reduction to their lifetime may arise). Therefore, unit commitment decisions, which must be taken some 24–48 hours in advance, must prepare the generation schedule in order to accommodate any forecasted drastic changes in wind power availability smoothly.

Going beyond research results, companies providing load forecasting services and tools advertise in the range of 1–3% of mean absolute percentage error (MAPE) for distribution load forecasting in short-term horizons [77]. From many published results, the range of MAPE for wind power short-term forecasting is in the range of 10–30% [78]. Therefore, we can say that the uncertainty in wind power prediction is about one order of magnitude larger than the uncertainty in load prediction. The latter is traditionally accommodated by defining a policy for spinning reserve (which may be some hybrid form of percentage of load plus some quantity related to the size of the largest generating unit in the system — to take finite reliability into account). However, with uncertainty in wind power prediction being much larger and extremely variable over the course of the hours of the day, the unit commitment exercise must consider alternative models that may take into account risk — because defining some sort of deterministic spinning reserve criterion may be too expensive, for the unused capacity or for shortages of capacity.

In order to allow unit commitment to take ramping into account, one must supply as input data information about the possibility of having a ramping event and about the probability of this event appearing shifted, relative to the predicted hour. This prediction cannot be performed without departing from a probabilistic model for the wind power prediction.

4.3.2.1 Generating a Discrete Representation of the Joint pdf of the Wind Power Prediction

The approach proposed in this chapter requires the existence of some wind power models, such that one is able to extract, in the Monte Carlo sense, sampled events within a given time horizon (see Fig. 4-4). Without loss of generality, we assume that the time horizon is of 48 hours. Therefore, a wind power scenario is any time series or sequence of wind power values within that time frame. This may also be seen as a point in a 48-h dimension space. Sampling wind power scenarios is thus sampling points in this 48-dimension space. A wind power prediction model that includes uncertainty representation has a way of generating scenarios according to the joint probability density function in this space.

One must bear in mind that in a wind power forecasting model, there are important cross-time dependencies, so that the prediction at each hour cannot be taken as an independent random variable. A traditional way to represent such dependencies is through second-order statistics by establishing correlation (or covariance) matrices. This approach is subject to comments and reservations; however, it is beyond the scope of this discussion. Nevertheless, it does not seem feasible to achieve an explicit mathematical formula describing the pdf of the prediction in the joint 48-dimension space.

One model that has been proposed in the literature [79], as applied to wind power forecasting, allows the generation of wind power scenarios from an implicit representation of the pdf via a set

of cross-time correlation matrices. Each scenario departs from a seed that is randomly sampled at the initial time step, and then the following time step values are generated with the help of the correlation matrices.

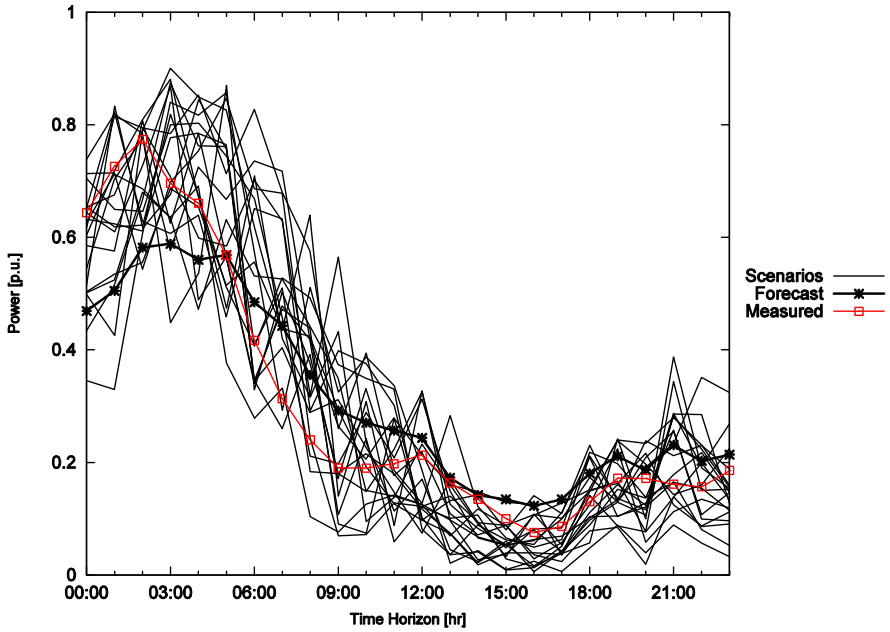

Fig. 4-4 Scenarios generated, point forecast, and actual measured value.

This chapter does not discuss the validity of this approach. However, this model allows one to sample and generate wind power scenarios in the Monte Carlo fashion. A scenario is an event, associated to which is a pdf value in the joint space. If one generates a large sample of scenarios, it is expected that these will spread in space and will be distributed with a frequency dictated by the pdf. A large sample will be a good discrete approximation of the true pdf. The availability of such a sample is the first step to take in the new model proposed.

4.3.2.2 Detecting Ramps

The second step is to have available a detection procedure for ramp events, such as those defined in the previous section. Given the industry's concerns, one needs an agreed-upon definition of a ramp event in order to proceed — this definition will be a typical wind power change that may be perceived as harmful by the industry. The definition is likely to vary, depending on the size of the region and the ramping flexibility in the portfolio of other generating resources, among other factors.

The simplest definition is in terms of slope. A general statement could be that given a time window of size T (with $T = n\,\Delta t$, n being an integer and Δt the size of the time step in the wind power series), a ramp is equivalent to a change $|\Delta P|$ in wind power above a certain threshold P_{ref}. Up or down is defined according to the sign of ΔP.

A ramp detection exercise may thus be carried out by running a moving window over a wind power series and applying a matching filter to detect similarities between any stretch of the time series and the ramp filter. Therefore, if this model is applied over one wind power scenario, it

indicates whether or not a matching ramping event is detected (see Fig. 4-1) for each hour at the beginning of the moving window.

4.3.2.3 Building a Probabilistic Ramp Representation

The detection model described above acts as an indicator function for each hour in the time series. The scenario generation model acts as a sampling mechanism in the Monte Carlo sense. The stage is set to generate a probabilistic representation of the ramp events implicit in the wind power prediction. In order to achieve such representation, the following sequence of operations must be followed:

1. Generate a large set of *N* wind power scenarios, sampled with the wind power forecasting model.
2. For each scenario, detect in each hour the possibility of having a ramp event of each of the types defined.
3. Count the total of ramp event detections in each hour for the whole sampled set of scenarios associated with each ramp type.
4. Based on the sample ratio of number of detected events n_i of type *i* over the size *N* of the sampled set, define probabilities for each ramp event in each hour of the forecasting horizon.

4.3.2.3.1 Event Occurrence

We define Vote Counting, V, for each time step *k*:

$$\mathrm{V_k} = \sum_{\mathrm{j=1}}^{\mathrm{N}} [\mathrm{F}\left(\Delta \mathrm{P_j^k}\right) > \mathrm{P_{ref}}] \tag{4-7}$$

where N is the number of predicted power signals/scenarios, F is one of the ramp definitions previously presented, ΔP is a variation in power, P_{ref} the admissible ramp magnitude, and $[X]$ a Boolean function, such that:

$$[\mathrm{X}] = \begin{cases} 0 & \text{if X = FALSE} \\ 1 & \text{if X = TRUE} \end{cases} \tag{4-8}$$

We can estimate the probability of an event, E, at a time step k with:

$$\mathrm{P(E_k)} = \frac{1}{\mathrm{N}} \mathrm{V_k} \tag{4-9}$$

By defining a cut-off threshold, *thr*, on the probability $P(E)$, we identify an event at time step *k* if $P(E_k) > thr$. In Section 4.4.1, we present a technique to find the optimum threshold. We use this methodology to identify either ramp-up or ramp-down events.

4.3.2.4 Building a Probabilistic Ramp Representation

Assume we have several scenarios for the wind power. All of the scenarios are for the same time horizon and granularity. We use histograms to aggregate ramp event detection, produced by several sources (scenarios), using any of the definitions presented in this chapter. As mentioned, each histogram interval corresponds to a specific variation in power; the value associated with the interval corresponds to a probability.

4.3.2.4.1 Building Histograms and Cumulative Ramp Probability Diagrams

To build a vertical histogram, we define a set of intervals $\Delta^b P$, ranging from P_{ref} to a user-specified maximum power change, and define the Vote Counting for each histogram interval b as:

$$V_k^b = \sum_{j=1}^{N} [\Delta^{bl} P < F(\Delta P_j^k) < \Delta^{bu} P] \quad (4\text{-}10)$$

where the lower and upper bound of the histogram intervals are bl and bu, respectively. These results allow one to build a cumulative diagram by adding the vote counting for all categories above each threshold. Fig. 4-5 illustrates the use that may be given to these cumulative ramp probability diagrams.

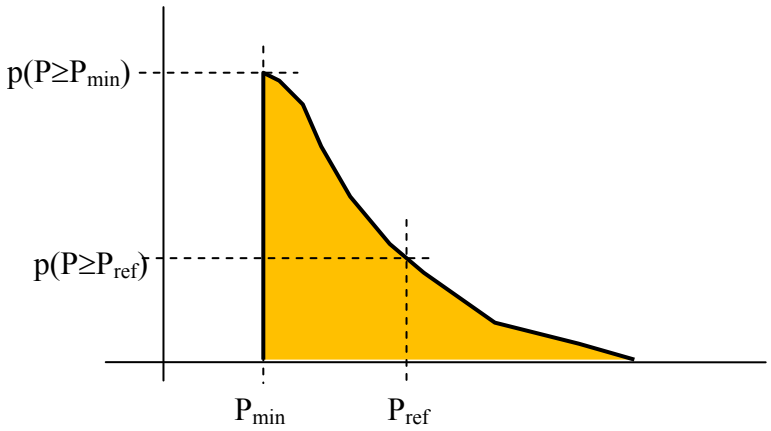

Fig. 4-5 Use of cumulative ramp probability diagrams. Given P_{ref} as a value in MW (or percentage of the wind farm nominal capacity), $p(P \geq P_{ref})$ gives the probability of having a ramp event with a change equal to or greater than P_{ref}. In this diagram, P_{min} represents the minimum value of power variation that is acceptable to trigger a ramp event alarm.

4.3.2.5 Visualization of Ramp Events

The following charts in Fig. 4-6, Fig. 4-7, and Fig. 4-8 present illustrative examples of using definition 5 with histograms to report ramps with different magnitudes and the corresponding probabilities. Blue histograms report Ramp-Up events; while red histograms report Ramp-Down events.

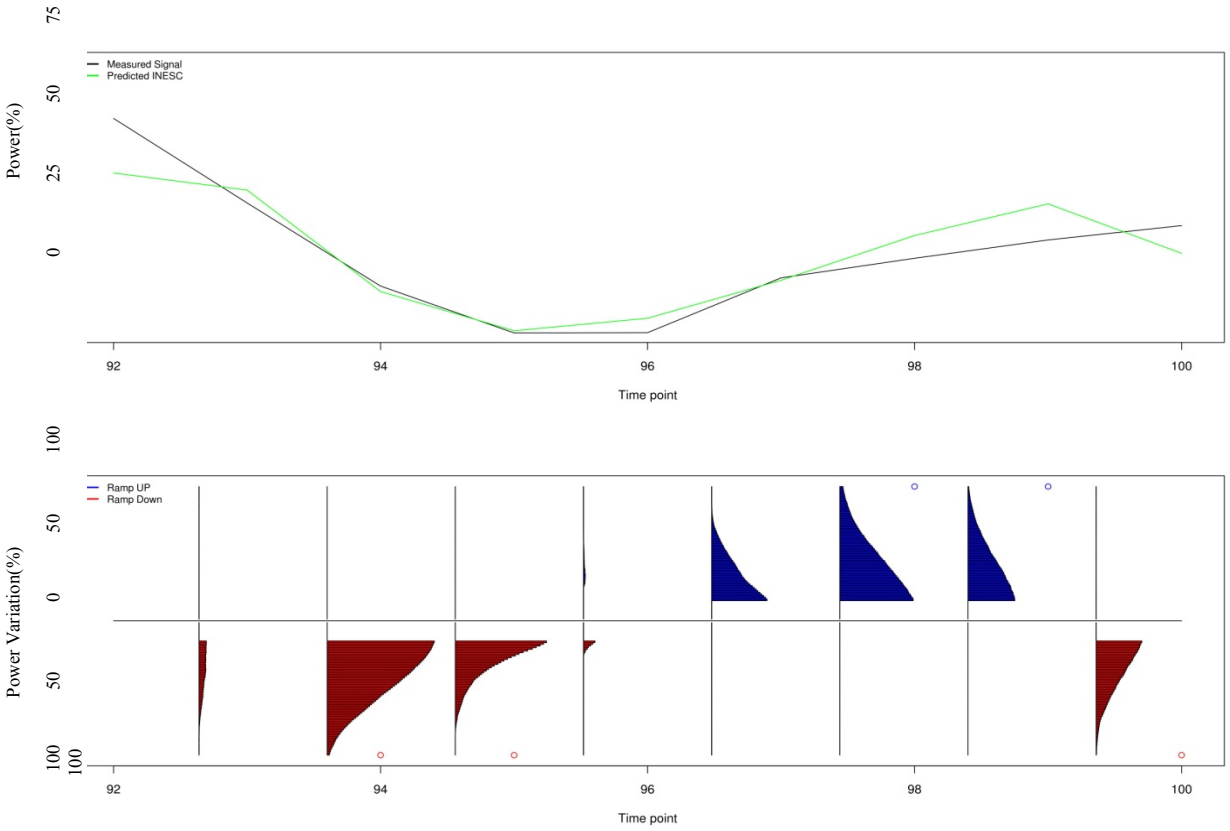

Fig. 4-6 Ramp-Up and Ramp-Down Histograms obtained using our voting method and definition 5. This figure also shows, in the subfigure above, one day of the wind farm production and the wind power point forecast for the same period. These results were obtained by using 3 hours' aggregation.

These figures, which are extracted from the case study data base detailed in the following chapter, present the point forecasting curve for three sampled days, as well as the actual wind power measured in the same period. These curves are indicative only, because the calculation is not based on them. The ramp probabilities are derived from a probabilistic representation of the wind power prediction. Because this model is based on scenarios obtained by a Monte Carlo sampling, the meaning of a ramp event probability being p is the following: p is the probability of following a wind power prediction scenario sampled for 24 hours that contains, at the given hour, a ramp event satisfying the definition.

Below these curves, there is a set of cumulative ramp probability diagrams for every three hours. The ramp definition encompassed 3 hours, and the results are displayed for 3-h time steps. However, the definition of ramp may be applied as a moving window every hour.

Notice that by inspecting the cumulative diagrams, in some hours there is a clear indication of ramp up or ramp down; in other cases, both movements may happen. Moreover, using the optimum threshold introduced in Section 4.3.2.3, in some hours the system identifies a ramp-up or ramp-down event by drawing circumferences. The system plots a blue circumference when a ramp-up event is identified and plots a red circunference when a ramp-down event is identified.

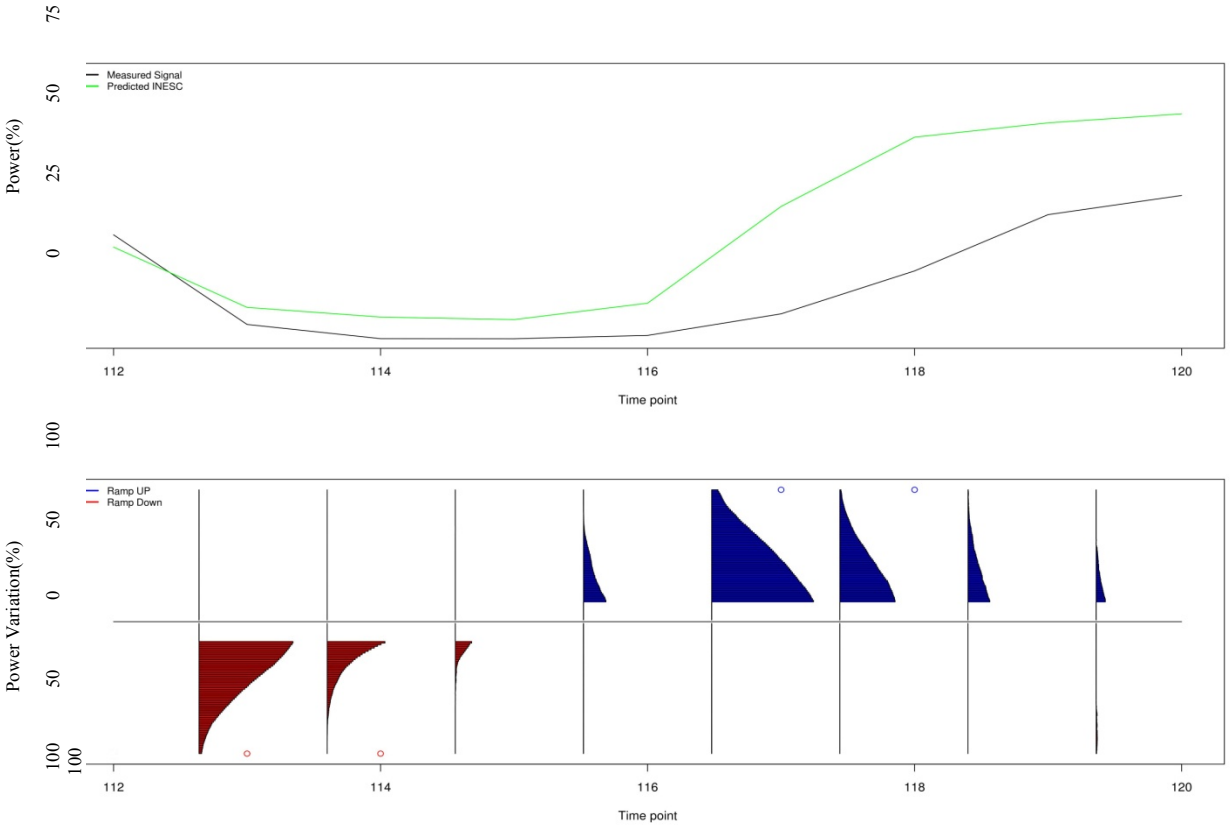

Fig. 4-7 Example of probability modeling of ramp-up and ramp-down possibilities.

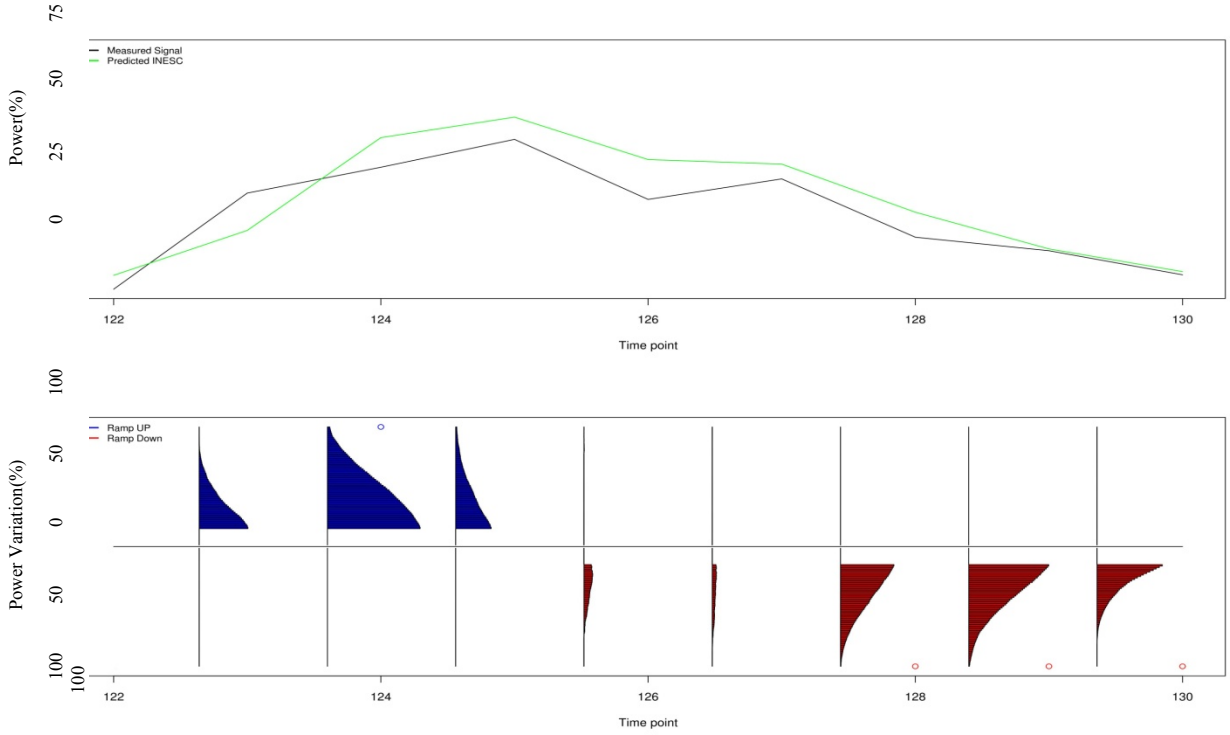

Fig. 4-8 Example of probability modeling of ramp-up and ramp-down possibilities.

4.4 Comparative Performance Assessment of the New Method

The system discussed in the previous section attaches a probability to the magnitude of ramp events. On the basis of this information, the decision maker decides whether or not to take action. For example, on the basis of the information that "there is an 80% probability of a ramp-up event to occur in the next 3 hours," the decision maker may or may not take an action. While the information is probabilistic, the action is not. The assessment of probabilistic forecasts must take into account the consequences, that is, the actions triggered by the information provided. Incorrect actions have associated costs, and different types of errors might have different costs. The cost associated with the action "act" when the event does not occur might be different from the cost of not having taken action when the event occurred. Depending on the costs (or the ratio of costs) of false alarms and misses, the decision maker might decide to act only when the probability of events is high. Key questions are: How do we define what constitutes a high probability? How do we define the decision threshold when costs are unknown? The Receiver Operating Characteristic (ROC) space allows us to plot a false alarm rate versus a missed rate, and find the "best" operating point under varying cost ratios [80][81].

In the following subsections, we present some metrics that can be used to evaluate ramp event forecasting systems. Furthermore, we will introduce phase error and the algorithm that we use to correct phase errors.

4.4.1 Metrics for Ramp Event Detection

Here we present metrics to evaluate event forecasters. First, we describe the metrics to evaluate deterministic forecast systems, and then we present metrics used to evaluate probabilistic forecasting systems, including basic concepts of the ROC space. It is important to note that the metrics that we present to evaluate deterministic forecasts can be used, by applying simple procedures, to evaluate probabilistic forecasting systems as well.

4.4.1.1 Determininstic Forecast

Two widely used statistics to evaluate the quality of deterministic forecast systems are Precision and Recall. Precision is defined as the ratio between the number of true positive events and the number of positive forecasts. Recall is defined as the ratio between the number of true positives and the number of observed positives. To illustrate the computation of these metrics, we present in Table 4-1 a general contingency table used to summarize the results of an event forecasting system.

Table 4-1 Contingency table representing event observation and event forecast.

	Yes	*No*
Yes	**TP** [a] *(hits)*	**FP** *(false alarms)*
No	**FN** *(misses)*	**TN**

[a] TP = true positive; FP = false positive; FN = false negative; TN = true negative.

Formally, and using the illustrative table, we can write the definition of True Positive Rate (TPR) as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{4-11}$$

Precision answers the question: What fraction of the predicted "yes" events really occur?

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}. \tag{4-12}$$

Recall answers the question: What fraction of the observed "yes" events were correctly forecast?

The F-Measure [82] combines Precision and Recall using the harmonic mean:

$$\text{F-Measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}} \tag{4-13}$$

In the context of ramp event detection, true negatives are irrelevant. Another useful metric is the Critical Success Index (CSI) [83], defined as:

$$\text{CSI} = \frac{\text{TP}}{\text{TP+FN+FP}} \tag{4-14}$$

Analogous to Precision and Recall, the CSI metric takes values in the interval [0;1], where 1 means correct prediction. CSI measures the fraction of observed and/or forecast events that were correctly predicted. It can be thought of as the accuracy achieved when correct negatives have been removed from consideration, that is, CSI is only concerned with forecasts that count. Being sensitive to hits, CSI penalizes both misses and false alarms. One of the works that uses such a metric to evaluate a ramp event forecast system is described in [73].

The Hanssen & Kuipper's Skill Score (KSS) [84][85], also known as Pierce's Skill Score or the True Skill Score, is a widely used metric that takes into account all of the elements of the contingency table. It measures the ability to separate "yes" events from the "no" events. The KSS can be defined by means of the hit rate $\left(H = \frac{TP}{TP+FN}\right)$ and false alarm rate $\left(F = \frac{FP}{FP+TN}\right)$ as:

$$\text{KSS} = \text{H} - \text{F} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{(\text{TP+FN})(\text{FP+TN})} \tag{4-15}$$

The KSS takes values in the interval [−1;1], where 0 indicates no skill and 1 the perfect score.

When predicting rare events having large TN values, the KSS approaches the value of the hit rate, thus becoming vulnerable to hedging, a strategy that consists in always forecasting event occurrences. This score is more appropriate for verifying frequently occurring events.

A special purpose score aiming to verify predictions of rare events is the Extreme Dependency Score (EDS) [86]. This metric does not account for the false positive alarms (FPs), nor the nonoccurring events (TNs); however, it considers the total number of cases for the sample size (n):

$$\mathrm{EDS} = \frac{2\log((\mathrm{TP}+\mathrm{FN})/\mathrm{n})}{\log(\mathrm{TP}/\mathrm{n})} - 1 \tag{4-16}$$

This metric has some good properties (as it does not tend to zero for vanishing events and it is not explicitly dependent on the bias), but it is sensible to hedging. The EDS score range is [−1,1], where −1 is the worst score and 1 is the perfect score. The EDS is −1 when the base rate $\left(BR = \frac{TP+FN}{n}\right)$ is one, and is one when the hit rate (H) is one. It answers the question: What is the relation between forecast and observed rare events?

Another metric that is suitable to analyse rare events, being less sensible to hedging, is the Odds Ratio (OR). This metric can be defined easily in terms of the hit rate (H) and false alarm rate (F):

$$\mathrm{OR} = \frac{\mathrm{H}/(1-\mathrm{H})}{\mathrm{F}/(1-\mathrm{F})} \tag{4-17}$$

The OR formula presents the ratio between the odds of making a hit and the odds of making a false alarm. The score of OR ranges from 0 to ∞. The perfect OR score is ∞, and the OR score exceeds one when the hit rate is higher than the false alarm rate. This measure answers the question: What is the ratio between the odds of a "yes" forecast and the odds of making a bad forecast?

4.4.1.2 Probabilistic Forecast

A probabilistic forecast assigns a probability to the prediction. A clear characteristic of probabilistic forecasts is the introduction of a degree of freedom: the use of a threshold on the probability to decide the ocurrence of events. A technique that can be used to assess the performance of a probabilistic forecast system and to choose the optimal threshold is the Receiver Operating Characteristic, or ROC curve. Another metric that can be used to assess the accuracy of a probabilistic forecast is the Brier Score [87].

4.4.1.2.1 The ROC Space

The ROC curve is a plot of the sensitivity versus the specificity of a binary classifier system, as its discrimination threshold is varied. The ROC curve is obtained by plotting the fraction of true positives (sensitivity) versus the fraction of false positives, as the criterion threshold changes [80][81].

The fraction of true positives was defined as Precison in (4-11). The fraction of false positives or the false positive rate is defined as

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \tag{4-18}$$

The best possible prediction method would yield a point in the upper left corner or coordinate (0;1) of the ROC space, representing a 100% sensitivity (no false negatives) and 100% specificity (no false positives). The diagonal of the ROC space corresponds to a random guess. A determinisc forecast defines a point in the ROC space. Points above the diagonal correspond to predictions better than a random choice, while points below correspond to predictions worse than

a random guess. A probabilistc forecast defines a trajectory in the ROC space, where each point in the trajectory corresponds to a different threshold. The threshold corresponding to the closest point to (0;1) is optimal in the sense that it provides the best trade-off between misses and false alarms.

Assume we have two vectors **P** and **O**, where **P** denotes the forecast vector and **O** denotes the vector of measurements. The vectors are of the same size, meaning that they refer to the same time interval and they are aligned; the same index in both vectors refers to the same time-stamp. All of the definitions reported in Section 4.2.2 can be rewritten in the form: $F(\Delta P, \Delta t) > P_{ref}$, where **ΔP** is a variation in power.

By applying one of the definitions previously described, we obtain two new vectors **ΔP** and **ΔO** that represent the variation in power given by the definition used. A negative value corresponds to reductions in power and are used to detect ramp downs, while a positive value corresponds to increases in power and thus is used in ramp-up detection.

Assume we know an admissible value of P_{ref}. The elements of **ΔO** are either above or below the reference power, denoting the occurrence of a ramp. Next, we sort both descending vectors according to the values in **ΔP**. It is expected that positive high values correspond to ramp-up events, while low negative values correspond to ramp-down events.

The ROC curve is generated as follows. Starting from the point (0,0) in the ROC space, we traverse top-down the **ΔO** vector (ordered descendant by the values in **ΔP**), if that point is above the reference, we move up by one step; otherwise, we move one step to the right. The 45° line corresponds to the best trade-off between the true positive rate and the false-positive rate. This line is the desired working point, assuming equal costs for both types of errors and uniform class distribution.

Fig. 4-9 illustrates the interpretation of the ROC space. A point on the trajectory with slope equal to 1, such as (FPR,TPR), is associated with the most convenient threshold to distinguish between detection and no-detection, when the cost of missing a positive is equal to the cost of assuming a positive when there is none. Other points with different slopes are associated with different cost relations. The ROC curve plots the performance of a forecaster for different operating points (i.e., for different and unknown cost ratios [81]).

More formally, consider that we are generating binary forecasts, where each observation can be labeled using one of two classes in the set {**yes**,**no**}, and a forecast can output the corresponding {**Yes**,**No**}(we use uppercase in the forecast labels for clarity). Consider that we know the distribution of **yes** and **no** events, that is, the probabilities $P(\mathbf{yes})$ and $P(\mathbf{no})$, and that we define the costs $\boldsymbol{cost}(\mathbf{Yes};\mathbf{no})$ and $\boldsymbol{cost}(\mathbf{No};\mathbf{yes})$ to be, respectively, the costs of predicting an event when no event occurs (a False Positive) and the cost of predicting no event when a event really occurs (a False Negative). Then, the slope of the line (a tangent line) that intersects the ROC curve at the optimum operating point, a point with coordinates (FPR_0, TPR_0) that is associated with a probability threshold, under the defined costs is:

$$\text{slope} = \frac{\text{P}(\mathbf{no}) \times \boldsymbol{cost}(\mathbf{Yes};\mathbf{no})}{\text{P}(\mathbf{yes}) \times \boldsymbol{cost}(\mathbf{No};\mathbf{yes})} \tag{4-19}$$

If we do not know this distribution, we can estimate the distribution from the observations.

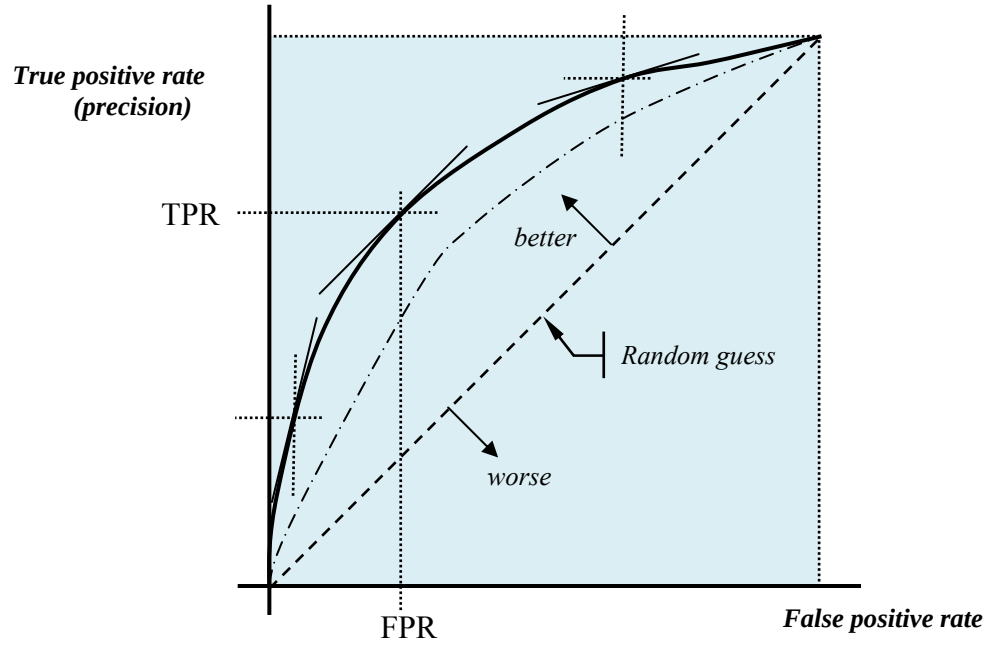

Fig. 4-9 Illustration of the ROC space. The solid curve describes the variation of (FPR, TPR) when the discriminating threshold that separates recognizing from not recognizing that an event occurred is changed. The dashed diagonal is the line associated with random guesses. The dash-dot curve corresponds to a different model, which is not as good as the one that produces the solid line.

The point (FPR_0, TPR_0) where the tangent line and the curve intersect is the optimum operating point, in the sense that this point minimizes the ***Expected Cost*** given by the following expression:

$$P(\mathbf{yes}) \times (1 - \text{TPR}) \times \boldsymbol{cost}(\mathbf{No};\mathbf{yes}) + P(\mathbf{no}) \times FPR \times \boldsymbol{cost}(\mathbf{Yes};\mathbf{no}) \tag{4-20}$$

If we define $P(\mathbf{Yes};\mathbf{no})$ as the probability of predicting an event when it does not occur (the probability of having a false positive) and $P(\mathbf{No};\mathbf{yes})$ the probability of predicting that an event does not occur when it really occurs (the probability of having a false negative), we can write the above formula as:

$$P(\mathbf{Yes};\mathbf{no}) \times \boldsymbol{cost}(\mathbf{Yes};\mathbf{no}) + P(\mathbf{No};\mathbf{yes}) \times \boldsymbol{cost}(\mathbf{No};\mathbf{yes}) \tag{4-21}$$

We obtain this formula by using the conditional probabilities and the equalities $TPR = P(\mathbf{Yes}|\mathbf{yes})$ and $FPR = P(\mathbf{Yes}|\mathbf{no})$.

Another useful method that relies on the ROC space and that can be used to select the best model/classifier from a set of models/classifiers is the ROC convex hull. The ROC convex hull is a ROC curve that connects the best operation points of a set of models/classifiers. By inspecting the ROC convex hull curve, we can choose the best model/classifier for specified class distributions and costs. Moreover, classifiers that are always bellow the ROC convex hull are considered suboptimal and can be discarded.

4.4.1.2.2 The Brier Score

For forecasts that assign a probabilty to each event, a more informed metric might be used. The Brier score is a score function that measures the accuracy of a set of probability assessments. It measures the average squared deviation between predicted probabilities for a set of events and their outcomes. It is computed as:

$$BS = \frac{1}{N}\sum_{t=1}^{N}(F_t - O_t)^2 \quad (4\text{-}22)$$

where F_t is the probability that was forecasted, O_t the actual outcome of the event at instance t (0 if it does not happen and 1 if it happens), and N is the number of forecasting instances. A lower score represents higher accuracy.

4.4.2 Phase Error

Errors in ramp event prediction can be split into two types: misses (or false negatives) and false alarms (or false positives). A ramp event occurred in the former, although the forecasting system does not predict the event. In this case, the output of a forecasting system may have time-shift predictions. If this behavior is generalized, we may say that the system predictions are affected by phase errors. This issue can be the result of a wide set of factors, including NWP errors, model bias, etc., and can severely affect the performance of some forecasting systems. In the latter case, there is no ramp event, and the forecasting system predicted an event.

To address the phase error issued in a forecasting system, we need to identify events that occur in a timestamp, t, not predicted at that time but predicted instead to occur in the time period immediately before or after the current one in the time interval $[t - \Delta t; t[\cup]t; t + \Delta t]$. The algorithm to correct phase errors starts by identifying each one of these shifted events. For each event, it updates the contingency table counts by adding 1 to true positives and decrementing both the number of false positives and the number of false negatives by one.

4.5 Experimental Evaluation

All of the definitions and methods presented in this chapter are mainly heuristic, and their assessment must be experimental. We may formulate several hypotheses: Is there any ramp definition that is better overall? Is there any metric that is better overall? The answer to these questions is no. In this section, we study the five definitions presented in Section 4.2.2 and present some interesting conclusions. We conclude that some definitions exhibit better performance than others under some circumstances. We start by describing the data and describing the experiments' configurations. Then, we present the results obtained. For each

metric, we present results for all parameters and for each type of ramp event. At the end of this chapter, we discuss the results and summarize the main findings.

4.5.1 The Data

In this study, we consider the data from the power measured in a large-scale wind farm located in the U.S. Midwest and 5,000 possible forecast scenarios (of wind power). This data correspond to a time period of 12 weeks (10/21/2009 to 02/18/2010). The data from the wind farm are measurements registered by a SCADA system that outputs measurements in a 10-minute time stamp, and the data from the scenarios are one-hour, time-stamp predictions generated by the wind power prediction platform developed in this project. The scenario generator launch time is 6 a.m., and all scenarios are generated for the 24 hours of the next day (i.e., the forecast horizon is from 18 hours to 42 hours).

4.5.2 Design of the Experiments

With these experiments, we aim to study and evaluate the performance of the probabilistic ramp detection system outlined above. Our system has a preprocessing stage, where we aggregate the measured and predicted signals, if needed, and two main components: the application of the ramp definitions and an event detection methodology, which is a voting scheme that outputs probabilities and that is used to forecast events.

In the preprocessing stage, we run experiments by aggregating the original signal using 1-, 2-, and 3-hours window aggregation. Concerning the study of ramp definitions, the second step consists in studying the sensitivity of each definition to two different parameters: the size of the time step (Δt) and phase error. For the size of the time step, we test Δt equal to 1, 2, 3, and 4 time periods. Regarding the nature of definitions four and five, as defined in Section 4.2.2, we do not analyze the sensitivity of these two definitions to the size of the time step. When analyzing definition five, we only present results for $\Delta t = 1$. For definition four, we analyze four values of the *nam* parameter, where *nam* equal to 1, 2, 3, and 4. Concerning the phase error, we analyze time-stamp errors of 0, 2, and 4 periods.

Concerning the forecast horizon, we do not present results for a specific time horizon. In our experiments, we compute all of the metrics by taking into account the full range of the scenarios forecast horizon, and we consider all events that occur from 18 hours to 42 hours ahead. Thus, we present what can be called a time-aggregated performance analysis.

Regarding the last phase of our system, we evaluate a probabilistic forecasting system that uses the voting scheme and a point forecasting system, both developed in this project. By presenting a comparison between the two systems, our goal is to analyze the advantages and disadvantages of using the probabilistic forecasting system. We hope to show that by exploring the degrees of freedom of the probability space, we can find the optimum probability threshold and make operating decisions under varying costs of misses or false alarms. Therefore, we compute a set of event detection metrics for a set of probability thresholds, ranging in the interval [0;1], and also present ROC curves. Moreover, we briefly present a decision-making framework that is grounded on ROC curves and considers equal and different costs for misses and false alarms.

Concerning the event detection metrics, we compute CSI, F-Measure, EDS, KSS, and Odds Ratio.

At this point, and before presenting the experimental results, we repeat the structure of our system. Assume we have two vectors **P** and **O**, where **P** denotes the forecast vector and **O** denotes the vector of measurements. By first applying a signal aggregation, we obtain two vectors of the same size that refer to the same time interval and are aligned; the same index in both vectors refers to the same time stamp. Then, all of the definitions reported in Section 4.2.2 can be rewritten in the form: $F(\Delta P, \Delta t) > P_{ref}$, where **ΔP** is a variation in power and one of the definitions previously described, and we obtain two new vectors **ΔP** and **ΔO** that represent the variation in power given by the definition used. A negative value corresponds to reductions in power and is used to detect ramp downs, whereas a positive value corresponds to increases in power and is being used in ramp-up detection.

Assume we know an admissible value of P_{ref}. The elements in **ΔP** and **ΔO** are either above or below the reference power, denoting the occurrence of a ramp. Then, by using this knowledge, we use the voting scheme described in Section 4.3.2.3.1 to obtain ramp probability at each time stamp. In the last phase, we use a probability threshold to identify event occurrence.

4.5.3 Experimental Results

In this subsection, we show tables and figures that summarize the results obtained by running our method using the data described in Section 4.5.1. We also present a comparison against a point forecast methodology.

First, we present results obtained by computing a set of metrics: CSI, F-Measure, KSS, EDS, and Odds Ratio; then, we present the ROC curves and some experiments that aim to demonstrate the contribution of ROC curves in decision support tasks. When analyzing each metric, we present results for both types of ramps (i.e., up and down ramps).

4.5.3.1 Experimental Results

Here we present the results obtained by running our system, a probabilistic forecasting system, a reference methodology, and a point forecasting system. To access the validity of our methodology, we compute a set of metrics: CSI, F-Measure, KSS, EDS, and Odds Ratio. For each metric, we present tables and figures for a specific setup and then discuss other main findings. Each table presents (1) the results for three-hour aggregation, defining an upward or downward ramp of magnitude change higher than 25% of the wind farm nominal power; and (2) the P_{ref} value, which occurs in a time period, which means $\Delta t = 1$. By using this Δt value, definitions one, two, and three lead to equal results. Regarding the nature of definition 4, a moving average, we present results by setting $nam = 2$. When setting $nam = 1$, we obtain almost the same results as those obtained when running definitions one, two, and three. Finally, we report results that show the advantage of including phase errors.

4.5.3.1.1 CSI

Table 4-2 presents comparative results between the probabilistic forecast and a point forecast system for the ramp-up event detection problem and the CSI metric. We compute the CSI metric for a set of probability thresholds, ranging in the interval [0;1]. The probability threshold that maximizes CSI is considered to be the optimum threshold. Probabilistic results are obtained using the optimum threshold, the point in Fig. 4-10 where CSI achieves the maximum value. In this example, the optimum threshold for definitions 4 and 5 was 0.2; and for definition 1, the best threshold was 0.3.

Table 4-2 Ramp-Up event detection and CSI metric value. Comparison between our methodology and a point forecast system.

RAMP UP – CSI						
	Probabilistic Forecast			Point Forecast		
	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
Events Occurred	63	101	77	63	101	77
Events Forecast	120	220	335	24	57	31
True Positives	20	53	55	7	23	8
True Negatives	813	708	619	896	841	876
False Positives	100	167	280	17	34	23
False Negatives	43	48	22	56	78	69
CSI	0.123	0.198	0.154	0.088	0.170	0.080

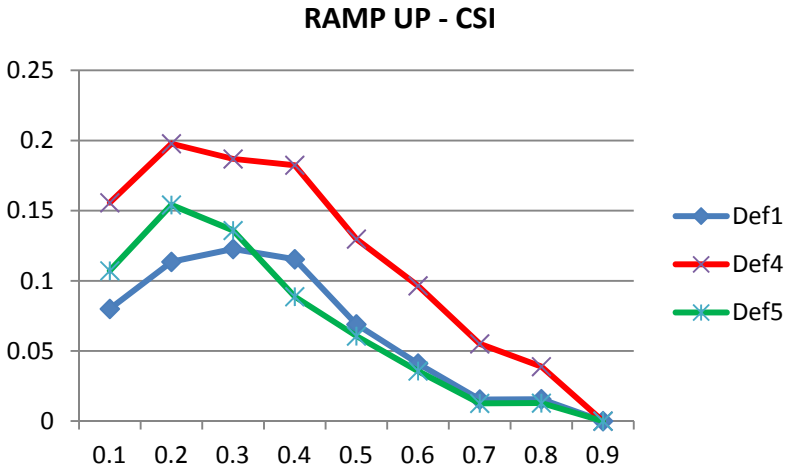

Fig. 4-10 CSI plot for different probability thresholds and ramp-up event detection. Better results correspond to high CSI values.

Table 4-3 presents comparative results between the probabilistic forecast and a point forecasting system for the ramp-down event detection problem and the CSI metric. Probabilistic results are obtained using the optimum threshold, the point in Fig. 4-11 where CSI achieves the maximum value. In this example, the optimum threshold for all three definitions was 0.3.

Table 4-3 Ramp-down event detection and CSI metric value. Comparison between our methodology and a point forecast system.

RAMP DOWN – CSI	Probabilistic Forecast			Point Forecast		
	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
Events Occurred	71	104	75	71	104	75
Events Forecast	116	137	103	28	61	32
True Positives	25	48	27	6	29	13
True Negatives	814	783	825	883	840	882
False Positives	91	89	76	22	32	19
False Negatives	46	56	48	65	75	62
CSI	0.154	0.249	0.179	0.065	0.213	0.138

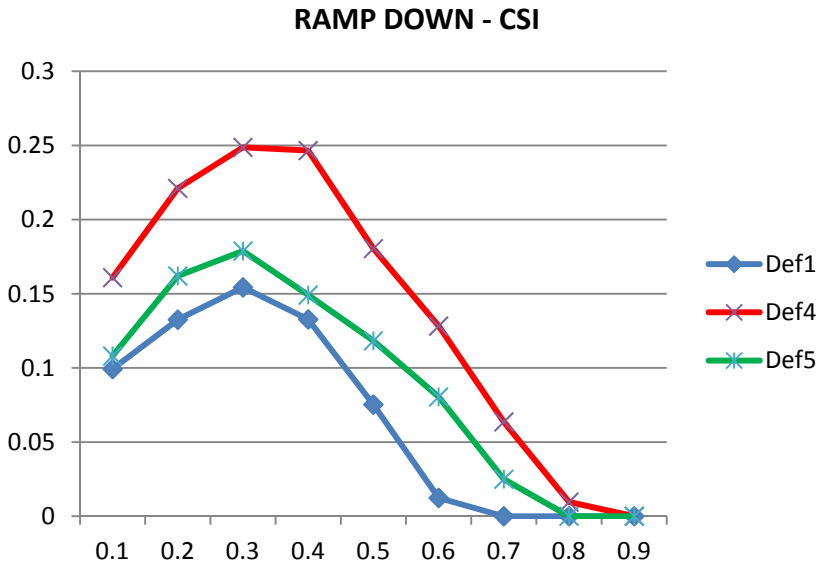

Fig. 4-11 CSI plot for different probability thresholds and ramp-down event detection. Better results correspond to high CSI values.

In Table 4-4 and Table 4-5, we present results by considering phase errors using the technique described in Section 4.1. As expected, by allowing phase errors of 2 and 4 time points, we reduce the FP number and obtain a high number of true positives, thus obtaining better CSI values.

Table 4-4 CSI taking phase error into account.

RAMP UP - CSI	Probabilistic Forecast			Point Forecast		
Phase Error (Δt)	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
0	0.123	0.198	0.154	0.088	0.170	0.080
2	0.303	0.358	0.323	0.176	0.306	0.241
4	0.381	0.451	0.375	0.261	0.374	0.317

Table 4-5 CSI taking phase error into account.

RAMP DOWN – CSI						
	Probabilistic Forecast			Point Forecast		
Phase Error (Δt)	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
0	0.154	0.249	0.179	0.065	0.213	0.138
2	0.333	0.362	0.299	0.193	0.310	0.189
4	0.403	0.452	0.391	0.207	0.352	0.230

Regarding the other parameters, the CSI metric increases with the aggregation in both ramp types. By looking at the size of the time step, Δt, we have two distinct cases. When addressing the ramp-up event detection problem, we obtain high CSI values for $\Delta t = 1$ and $\Delta t = 2$, and lower values for $\Delta t = 3$ and $\Delta t = 4$. Conversely, we obtain high CSI values for $\Delta t = 3$ and $\Delta t = 4$, and low values for $\Delta t = 1$ and $\Delta t = 2$.

Moreover, if we inspect the results obtained by running definitions 2 and 3, we observe that, for definition 2, the CSI value increases with the size of the time step and, for $\Delta t > 1$, we obtain better CSI results than the ones obtained by using definition 1. On the other hand, by using definition 3, we obtain worse results with the increase of the size of the time step.

By looking at the results obtained by using definition four, we can say that by setting the *nam* parameter equal to one, we obtain almost the same results as the ones presented by definitions 1, 2, 3, and 5. When we increase the value of the *nam* parameter, we obtain better CSI values for definition 4 than for the first three definitions, even if we set the size of the time step equal to the *nam* parameter value. As already mentioned above, we do not introduce the size of the time step in definition 5, and the results that we obtain by running this definition only show that CSI values increase with the aggregation size and the phase error. This conclusion is also valid for all of the other four definitions. We can also say that we obtain higher CSI values when running our system to forecast ramp-down events than when we run our system to identify ramp-up events. This result is especially clear when we use large sizes of the time step.

Furthermore, we can say that, despite the parameters configuration and ramp types, the optimum probability threshold is usually found within the interval $[0.1; 0.4]$.

4.5.3.1.2 F- Measure

Table 4-6 presents comparative results between the probabilistic forecast and a point forecast system for the ramp-up event detection problem and the F-Measure metric. Probabilistic results are obtained using the optimum threshold, the point in Fig. 4-12 where F-Measure achieves the maximum value. In this example, the optimum threshold for definitions 4 and 5 was 0.2, and for definition 1 the best threshold was 0.3. This result is quite similar to the one obtained by computing the CSI metric.

Table 4-6 Ramp-Up event detection and F-Measure metric value. Comparison between our methodology and a point forecast system.

RAMP UP – F-Measure	Probabilistic Forecast			Point Forecast		
	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
Events Occurred	63	101	77	63	101	77
Events Forecast	120	220	335	24	57	31
True Positives	20	53	55	7	23	8
True Negatives	813	708	619	896	841	876
False Positives	100	167	280	17	34	23
False Negatives	43	48	22	56	78	69
F-Measure	0.219	0.330	0.267	0.161	0.291	0.149

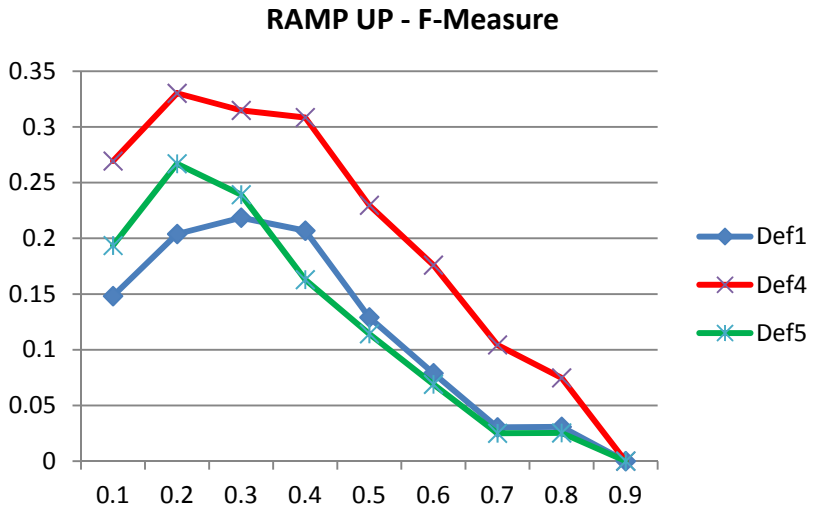

Fig. 4-12 F-Measure plot for different probability thresholds and ramp-up event detection. Better results correspond to high F-Measure values.

Table 4-7 presents comparative results between the probabilistic forecast and a point forecast system for the ramp-down event detection problem and F-Measure metric. Probabilistic results are obtained using the optimum threshold, the point in Fig. 4-13 where F-Measure achieves the maximum value. In this example, the optimum threshold for definitions 1 and 5 was 0.3, and for definition 4, it was 0.4.

Table 4-7 Ramp-Down event detection and F-Measure metric value. Comparison between our methodology and a point forecast system.

RAMP DOWN – F-Measure						
	Probabilistic Forecast			Point Forecast		
	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
Events Occurred	71	104	75	71	104	75
Events Forecast	117	137	103	28	61	32
True Positives	25	48	27	6	29	13
True Negatives	813	783	825	883	840	882
False Positives	92	89	76	22	32	19
False Negatives	46	56	48	65	75	62
F-Measure	0.266	0.399	0.303	0.121	0.352	0.243

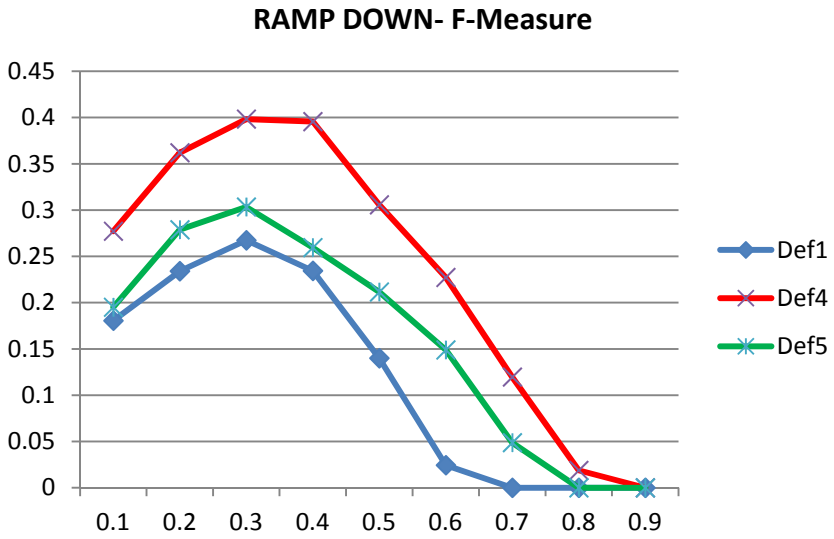

Fig. 4-13 F-Measure plot for different probability thresholds and ramp-down event detection. Better results correspond to high F-Measure values.

In Table 4-8 and Table 4-9, we present results by allowing phase errors. As expected, by allowing phase errors of 2 and 4 time points, we reduce the false positive number and obtain a high number of true positives, thus obtaining better F-Measure values. Inside brackets, and in small font, we present, for each definition, the ratio between the F-Measure value of the current cell and the F-Measure value obtained by not considering any phase error, $\Delta t = 0$.

Table 4-8 F-Measure taking phase error into account.

RAMP UP – F-Measure						
	Probabilistic Forecast			Point Forecast		
Phase Error (Δt)	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
0	0.219	0.330	0.267	0.161	0.291	0.149
2	0.303	0.528	0.488	0.299	0.468	0.389
4	0.552	0.621	0.545	0.414	0.544	0.481

Table 4-9 F-Measure taking phase error into account.

RAMP DOWN – F-Measure						
	Probabilistic Forecast			Point Forecast		
Phase Error (Δt)	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
0	0.266	0.399	0.303	0.121	0.352	0.243
2	0.500	0.531	0.461	0.323	0.473	0.318
4	0.574	0.622	0.562	0.343	0.521	0.374

By inspecting the F-Measure values for all results, the conclusions are the same as the ones when computing the CSI metric. There are no significant advantages of one metric over the other.

4.5.3.1.3 EDS

Table 4-10 presents comparative results between the probabilistic forecast and a point forecast system for the ramp-up event detection problem and the EDS metric. Probabilistic results are obtained using the optimum threshold, the point in Fig. 4-14 where EDS achieves the maximum value. In this example, the optimum threshold for all three definitions was 0.1.

Table 4-10 Ramp-UP event detection and EDS metric value. Comparison between our methodology and a point forecast system.

RAMP UP – EDS						
	Probabilistic Forecast			Point Forecast		
	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
Events Occurred	63	101	77	63	101	77
Events Forecast	666	530	636	24	57	31
True Positives	54	85	69	7	23	8
True Negatives	301	430	332	896	841	876
False Positives	612	445	567	17	34	23
False Negatives	9	16	8	56	78	69
EDS	0.893	0.859	0.917	0.110	0.210	0.057

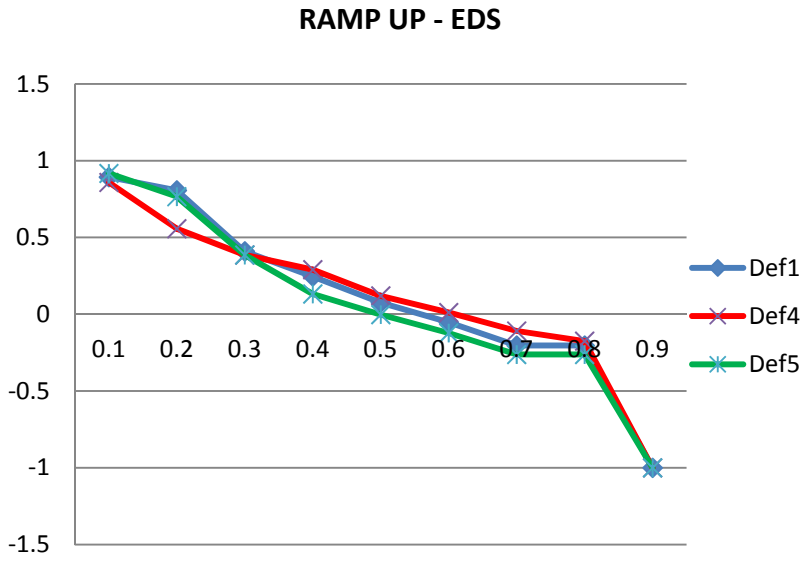

Fig. 4-14 EDS plot for different probability thresholds and ramp-up event detection. Better results correspond to high EDS values.

Table 4-11 presents comparative results between the probabilistic forecast and a point forecast system for the ramp-down event detection problem and the EDS metric. Probabilistic results are obtained using the optimum threshold, the point in Fig. 4-15 where EDS achieves the maximum value. In this example, the optimum threshold for all definitions was 0.1.

Table 4-11 Ramp-Down event detection and EDS metric value. Comparison between our methodology and a point forecast system.

RAMP DOWN – EDS						
	Probabilistic Forecast			Point Forecast		
	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
Events Occurred	71	104	75	71	104	75
Events Forecast	661	531	612	28	61	32
True Positives	66	88	67	6	29	13
True Negatives	310	429	356	883	840	882
False Positives	595	443	545	22	32	19
False Negatives	5	16	8	65	75	62
EDS	0.945	0.861	0.916	0.029	0.274	0.188

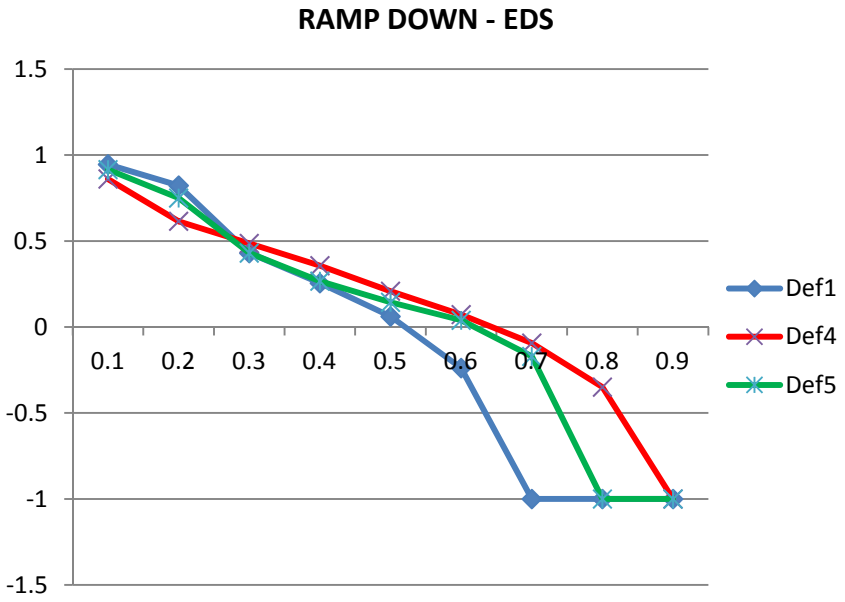

Fig. 4-15 EDS plot for different probability thresholds and ramp-down event detection. Better results correspond to high EDS values.

In Table 4-12 and Table 4-13, we present results by allowing phase errors. By allowing phase errors of 2 and 4 time points, we reduced the false positive number and obtained a high number of true positives, thus obtaining better EDS values.

Table 4-12 EDS taking phase error into account.

RAMP UP – EDS

	Probabilistic Forecast			Point Forecast		
Phase Error (Δt)	**DEF1**	**DEF4**	**DEF5**	**DEF1**	**DEF4**	**DEF5**
0	0.893	0.859	0.917	0.110	0.210	0.057
2	1	0.991	0.990	0.270	0.386	0.323
4	1	0.991	1	0.373	0.453	0.401

Table 4-13 EDS taking phase error into account.

RAMP DOWN – EDS

	Probabilistic Forecast			Point Forecast		
Phase Error (Δt)	**DEF1**	**DEF4**	**DEF5**	**DEF1**	**DEF4**	**DEF5**
0	0.945	0.861	0.916	0.029	0.274	0.188
2	1	0.966	0.979	0.275	0.391	0.267
4	1	0.991	1	0.294	0.434	0.320

By looking at EDS results, we can say that the EDS values decrease with the size of the time step, when running definition 1, and increase when running definition 2. Regarding definition 4, we obtain worse results with the decrease of the *nam* parameter. In general, for all definitions, we obtain better results for low probability thresholds and large aggregations. As expected, we also obtain better results by considering phase errors.

As in the previous two metrics, when we run our system to identify ramp-down events, we obtain higher EDS values than when we run our algorithm to identify ramp-up events.

4.5.3.1.4 OR

Table 4-14 presents comparative results between the probabilistic forecast and a point forecast system for the ramp-up event detection problem and OR metric. Probabilistic results are obtained using the optimum threshold, the point in Fig. 4-16 where OR achieves the maximum value. In this example, the optimum threshold for all three definitions was 0.8.

Table 4-14 Ramp-UP event detection and OR metric value. Comparison between our methodology and a point forecast system.

RAMP UP – OR	Probabilistic Forecast			Point Forecast		
	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
Events Occurred	63	101	77	63	101	77
Events Forecast	2	6	2	24	57	31
True Positives	1	4	1	7	23	8
True Negatives	912	873	898	896	841	876
False Positives	612	2	1	17	34	23
False Negatives	1	97	76	56	78	69
OR	14.710	18	11.816	6,588	7.294	4.416

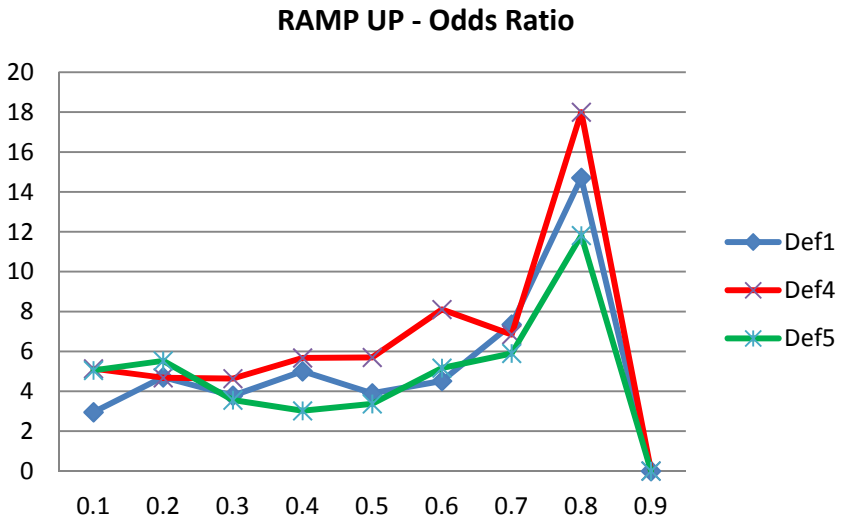

Fig. 4-16 Odds Ratio plot for different probability thresholds and ramp-up event detection. Better results correspond to high CSI values.

Table 4-15 presents comparative results between the probabilistic forecast and a point forecast system for the ramp-down event detection problem and OR metric. Probabilistic results are obtained using the optimum threshold, the point in Fig. 4-17 where OR achieves the maximum value. In this example, the optimum threshold for definition 1 was 0.1, for definition 4 was 0.6, and for definition 5 was 0.5.

Table 4-15 Ramp-Down event detection and OR metric value. Comparison between our methodology and a point forecast system.

RAMP DOWN – OR						
	Probabilistic Forecast			Point Forecast		
	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
Events Occurred	71	104	75	71	104	75
Events Forecast	661	28	29	28	61	32
True Positives	66	15	11	6	29	13
True Negatives	310	859	883	883	840	882
False Positives	595	13	18	22	32	19
False Negatives	5	89	64	65	75	62
OR	6.877	11.137	8.431	3.705	10.150	9.733

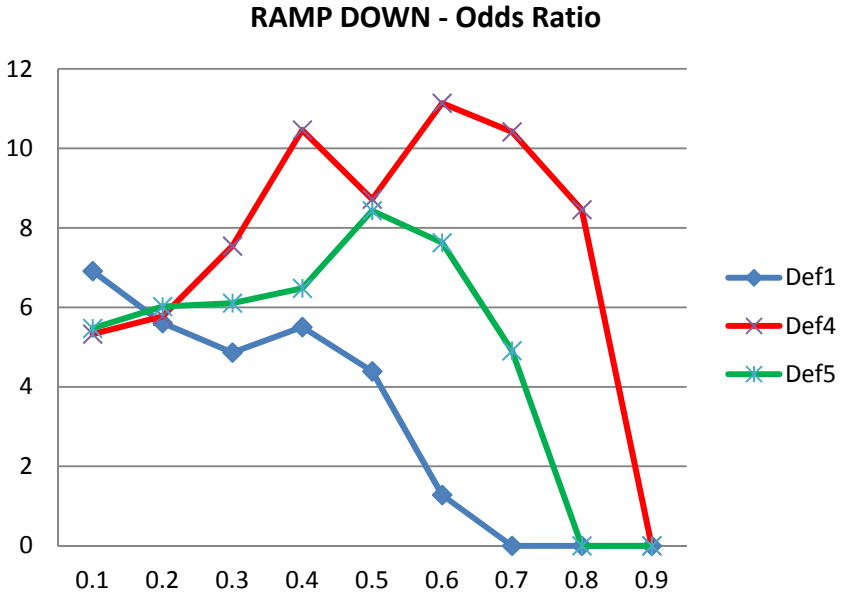

Fig. 4-17 Odds Ratio plot for different probability thresholds and ramp-down event detection. Better results correspond to high Odds Ratio values.

In Table 4-16 and Table 4-17, we present results by allowing phase errors. By allowing phase errors of 2 and 4 time points, we reduce the false positive number and obtain a high number of true positives, thus obtaining better OR values. In side brackets and in the small font, we present,

for each definition, the ratio between the OR value of the current cell and the OR value obtained by not considering any phase error, $\Delta t = 0$.

Table 4-16 OR taking phase error into account.

RAMP UP – OR						
	Probabilistic Forecast			Point Forecast		
Phase Error (Δt)	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
0	14.710	18	11.816	6.588	7.294	4.416
2	∞	103.488	∞	21.32	24.715	33.338
4	∞	103.488	∞	60.467	45.595	91.152

Table 4-17 OR taking phase error into account.

RAMP DOWN – OR						
	Probabilistic Forecast			Point Forecast		
Phase Error (Δt)	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
0	6.877	11.137	8.431	3.705	10.150	9.733
2	∞	30.816	34.747	21.648	23.182	17.313
4	∞	106.850	∞	25.586	33.444	26.940

When evaluating the OR results, we can say that the OR values decrease with increasing time step size, either when running our system to identify ramp-up or ramp-down events. Strangely, when running definition 4, the OR values increase with the *nam* parameter. We also obtained higher OR values when running our system to forecast ramp-up events than we did when running it to forecast ramp-down events. Another interesting issue is the shape of the OR curve. When we set the size of the time step to small values, we obtain the maximum OR values for large probability thresholds. When we work with larger time steps, a parabola shape results; the maximum OR values are near the probabilities 0 and 1. Again, the OR increases with aggregation and phase error.

4.5.3.1.5 KSS

Table 4-18 presents comparative results between the probabilistic forecast and a point forecast system for the ramp-up event detection problem and KSS metric. Probabilistic results are obtained using the optimum threshold, the point in Fig. 4-18 where KSS achieves the maximum value. In this example, the optimum threshold for all definitions was 0.2.

Table 4-18 Ramp-UP event detection and KSS metric value. Comparison between our methodology and a point forecast system.

RAMP UP – KSS	Probabilistic Forecast			Point Forecast		
	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
Events Occurred	63	101	77	63	101	77
Events Forecast	398	220	335	24	57	31
True Positives	47	53	55	7	23	8
True Negatives	562	708	619	896	841	876
False Positives	351	167	280	17	34	23
False Negatives	16	48	22	56	78	69
KSS	0.362	0.334	0.403	0.092	0.189	0.078

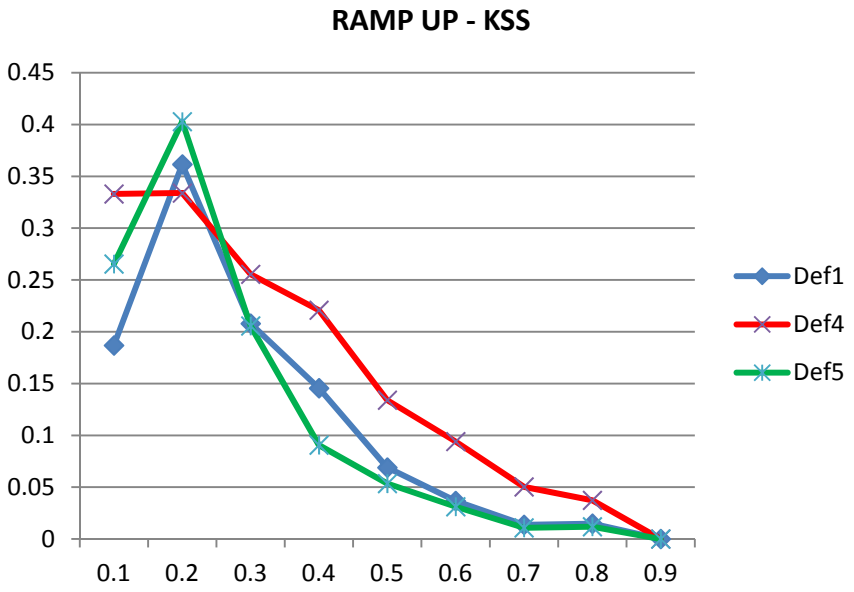

Fig. 4-18 KSS plot for different probability thresholds and ramp-up event detection. Better results correspond to high KSS values.

Table 4-19 presents comparative results between the probabilistic forecast and a point forecast system for the ramp-down event detection problem and KSS metric. Probabilistic results are obtained using the optimum threshold, the point in Fig. 4-19 where KSS achieves the maximum value. In this example, the optimum threshold for all definitions was 0.2.

Table 4-19 Ramp-Down event detection and KSS metric value. Comparison between our methodology and a point forecast system.

RAMP DOWN – KSS	Probabilistic Forecast			Point Forecast		
	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
Events Occurred	71	104	75	71	104	75
Events Forecast	400	233	298	28	61	32
True Positives	55	61	52	6	29	13
True Negatives	560	700	655	883	840	882
False Positives	345	172	246	22	32	19
False Negatives	16	43	23	65	75	62
KSS	0.393	0.389	0.420	0.060	0.242	0.152

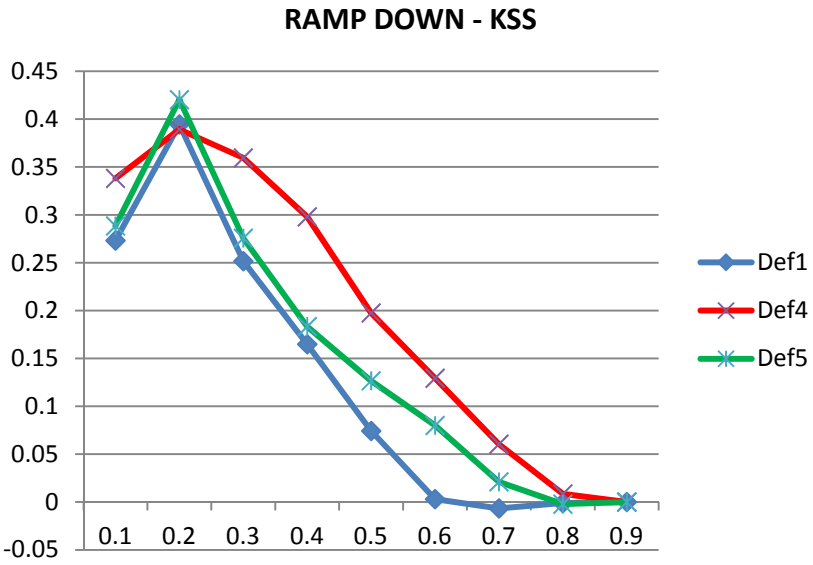

Fig. 4-19 KSS plot for different probability thresholds and ramp-down event detection. Better results correspond to high KSS values.

In Table 4-20 and Table 4-21, we present results by allowing phase errors. By allowing phase errors of 2 and 4 time points, we reduce the false positive number and obtain a high number of true positives, thus obtaining better KSS values. In side brackets and in small font, we present, for each definition, the ratio between the KSS value of the current cell and the KSS value that we obtain by not considering any phase error, $\Delta t = 0$.

Table 4-20 KSS taking phase error into account.

RAMP UP – KSS						
	Probabilistic Forecast			Point Forecast		
Phase Error (Δt)	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
0	0.362	0.334	0.403	0.092	0.189	0.078
2	0.599	0.632	0.685	0.194	0.343	0.262
4	0.666	0.754	0.713	0.280	0.410	0.332

Table 4-21 KSS taking phase error into account.

RAMP DOWN – KSS						
	Probabilistic Forecast			Point Forecast		
Phase Error (Δt)	DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
0	0.393	0.389	0.420	0.060	0.242	0.152
2	0.606	0.648	0.666	0.212	0.350	0.210
4	0.691	0.744	0.724	0.227	0.393	0.253

The results obtained by computing the KSS metric, to access the performance of our probabilistic system, is similar to the ones that we obtained when computing the EDS metrics. We get almost the same behavior and there is no relevant information to add here.

4.5.3.1.6 ROC Curves

Fig. 4-20 presents the ROC curves for definitions 1, 4, and 5, for different P_{ref} values. The best-performing curves are those that define the convex hull. For different working regimes (the ratio of costs between false alarm rate and false negative rate), different P_{ref} values must be chosen.

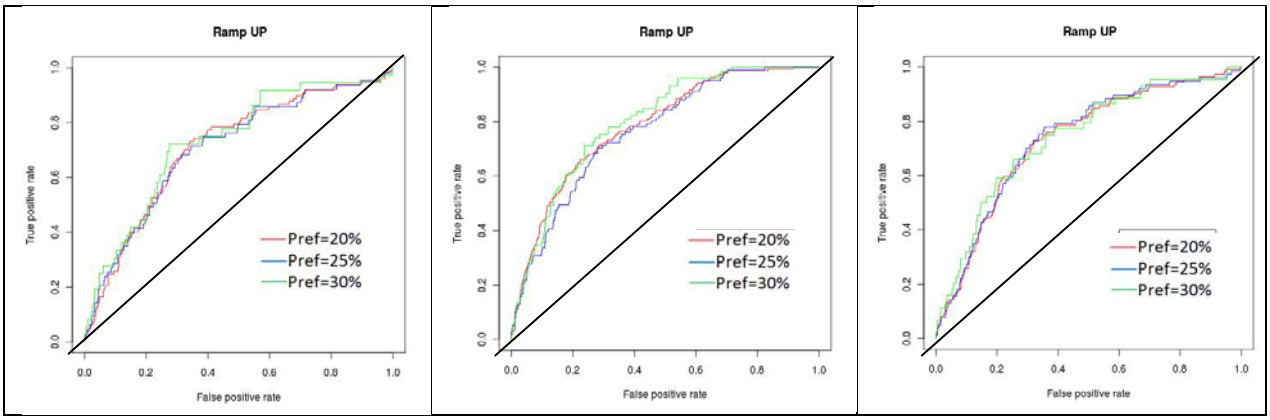

Fig. 4-20 ROC Curves for definitions 1, 4, and 5, respectively, for different P_{ref} values (20%, 25%, and 30% power change).

Fig. 4-21 plots the best ROC curve for different ramp definitions. The convex hull is defined by the curves corresponding to definitions 4 and 5, which indicate some advantages of using these definitions to detect ramp ups.

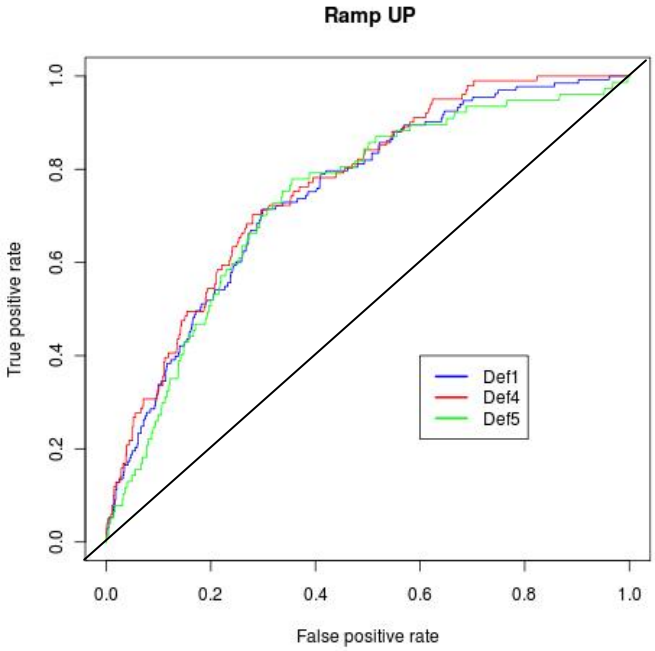

Fig. 4-21 ROC curves for the different methods under evaluation (25% power change).

Fig. 4-22 presents the ROC Curves for definitions 1, 4, and 5 for different P_{ref} values.

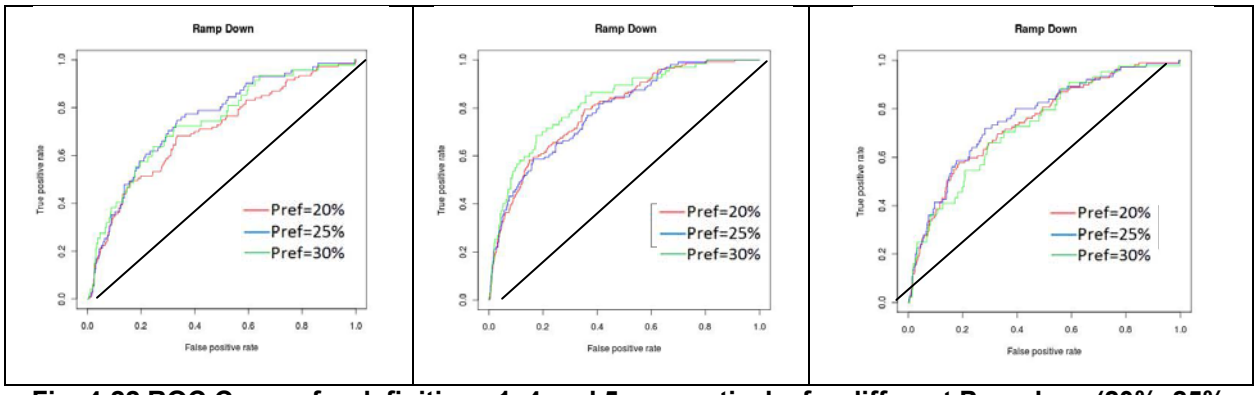

Fig. 4-22 ROC Curves for definitions 1, 4 and 5, respectively, for different P_{ref} values (20%, 25% and 30% power change).

Fig. 4-23 plots the best ROC curve for different ramp definitions. Again, the convex hull is defined by the curves corresponding to definitions 4 and 5, which indicate some advantages of using these definitions to detect ramp downs.

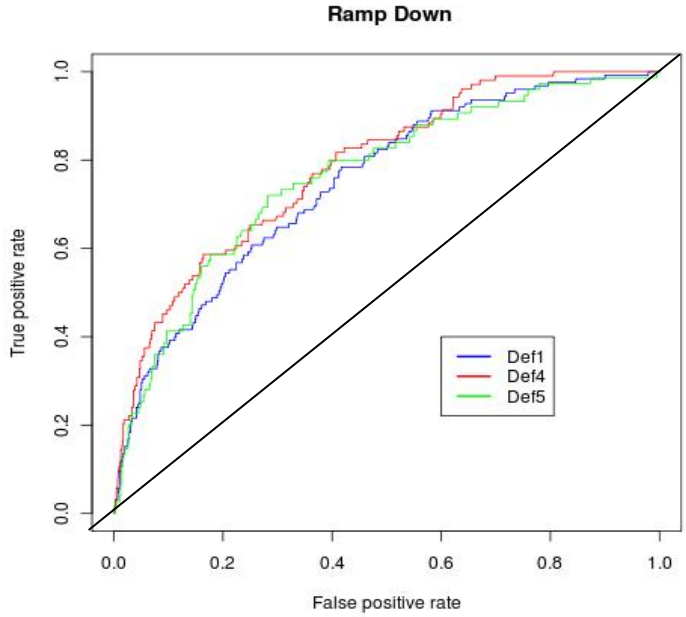

Fig. 4-23 ROC curves for the different methods under evaluation (25% power change).

4.5.3.1.6.1 Optimum Operating Point and Minimizing the Expected Cost

In this subsection, we present the optimum operating point for two cost configurations, each one considering the cost of False Positives and the cost of False Negatives. In configuration one, we define the error costs to be $\boldsymbol{cost}(\mathbf{No};\mathbf{yes}) = 200$, that is, the cost of an FN (cFN), and $\boldsymbol{cost}(\mathbf{Yes};\mathbf{no}) = 10$, that is, the cost of an FP (cFP). In configuration two, we consider $\boldsymbol{cost}(\mathbf{No};\mathbf{yes}) = 10$ and $\boldsymbol{cost}(\mathbf{Yes};\mathbf{no}) = 200$. To compute the slope of the tangent line and the Expected Cost, we use equations (4-19) and (4-21) given in Section 4.4.1.2.1.

We first present results showing the optimal operating point for both types of ramps, and then we present the figures, analyzing the probabilistic forecasting costs. The results presented in each table or figure use the same settings described above: a three-hour aggregation, where the size of the time step is equal to one and phase errors are not considered.

In Table 4-22, for each ramp type and three definitions, we present the number of time points, the number of events that occurred (Yes events), and the number of non-events (No events), as well as two slopes — one for configuration one and another for configuration two.

Table 4-22 Slope of tangent line for the two cost configurations and both ramp types.

		Slope of the Tangent Line					
		RAMP UP			RAMP DOWN		
		DEF1	DEF4	DEF5	DEF1	DEF4	DEF5
	Time points	976	976	976	976	976	976
	Number of Yes events	63	101	77	71	104	75
	Number of No events	913	875	899	905	872	901
Slope	Conf. 1 (cFN=200;cFP=10)	0,725	0,433	0,584	0,637	0,419	0,601
	Conf. 2 (cFN=10;cFP=200)	289,841	173,267	233,506	254,930	167,692	240,267

In Fig. 4-24 and Fig. 4-25, we present the ROC curves obtained when predicting upward and downward ramps, respectively, using definitions 1, 4, and 5. In these ROC curves, we can see some probability thresholds associated with a point (FPR, TPR). We also present two lines, each associated with a cost configuration. The blue line is the tangent line at the optimum point according to configuration one; the red line is the tangent line according to configuration two.

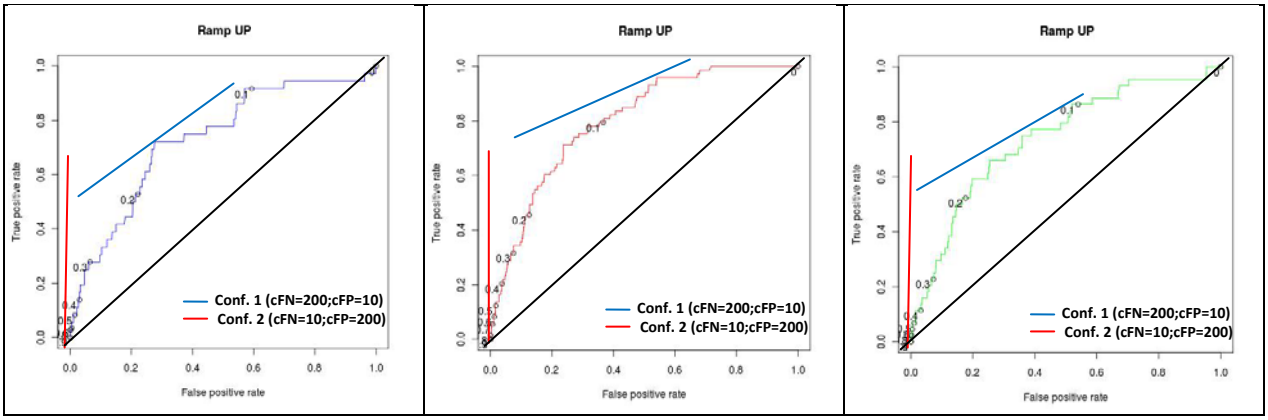

Fig. 4-24 ROC curves for ramp-up event detection using the definitions 1, 4, and 5, respectively.

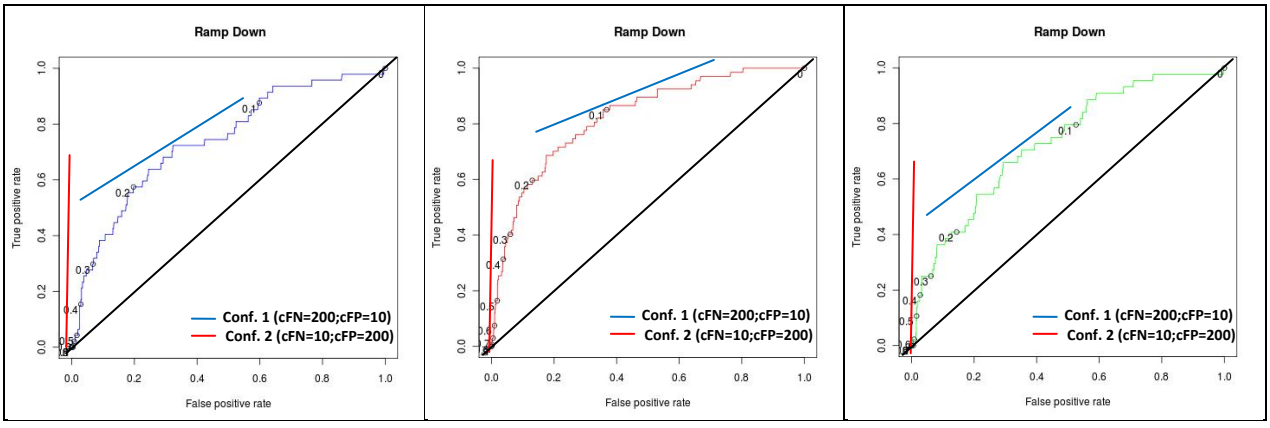

Fig. 4-25 ROC curves for ramp-down event detection using the definitions 1, 4, and 5, respectively.

By computing the slope and drawing the ROC curves and tangent lines, we can obtain the optimum operating point, according to the user-defined costs; however, we cannot directly access the expected costs for the full range of probability thresholds.

In Fig. 4-26, Fig. 4-27, and Fig. 4-28, we present, for the ramp-up events, the expected cost (see Equation [4-21]) for each definition as a function of cut-off frequency. In each figure, we present the expected cost for each of the two cost configurations. By using these figures, we can easily identify the optimum operating point, the one that was identified above when using the tangent line. This kind of analysis has some advantages over the previous one, as we can directly obtain the expected cost for each probability threshold. In Fig. 4-29, Fig. 4-30, and Fig. 4-31, we present the same information for the ramp-down events.

By inspecting the figures, we can see that for configuration one (that penalizes the FN severely), we obtain the optimum operating threshold, ranging from 0.0 to 0.2. The optimum operating threshold is the probability threshold where we obtain the minimum expected cost. This result means that we need a small percentage of scenarios voting to forecast a ramp; in another way, we will predict a ramp in a wide number of points, given that FPs are penalized less than are the FNs. In contrast, when we analyze configuration two, we see the expected cost decreasing when the probability threshold increases. This result means that a ramp event will be hard to predict because we need a large percentage of scenarios saying that an event will occur. This finding means that we need to be more careful when predicting a ramp event, given that FPs are severely penalized.

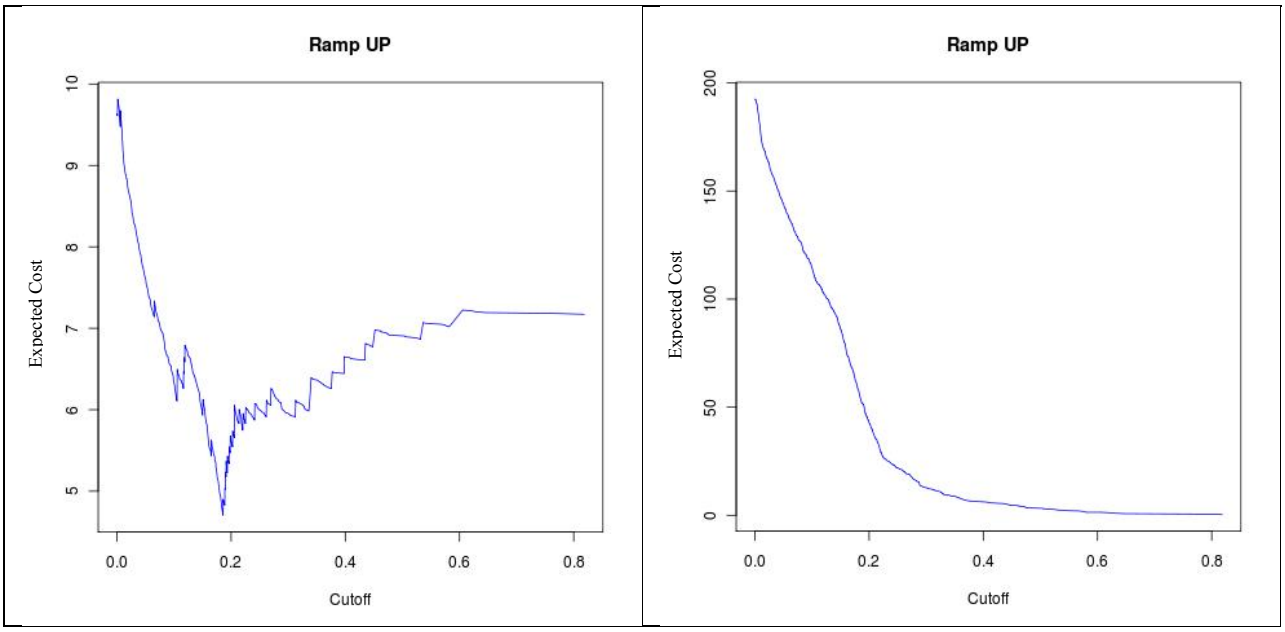

Fig. 4-26 Expected cost using definition 1: cFN=200 and cFP=10 (left), and cFN=10 and cFP=200 (right).

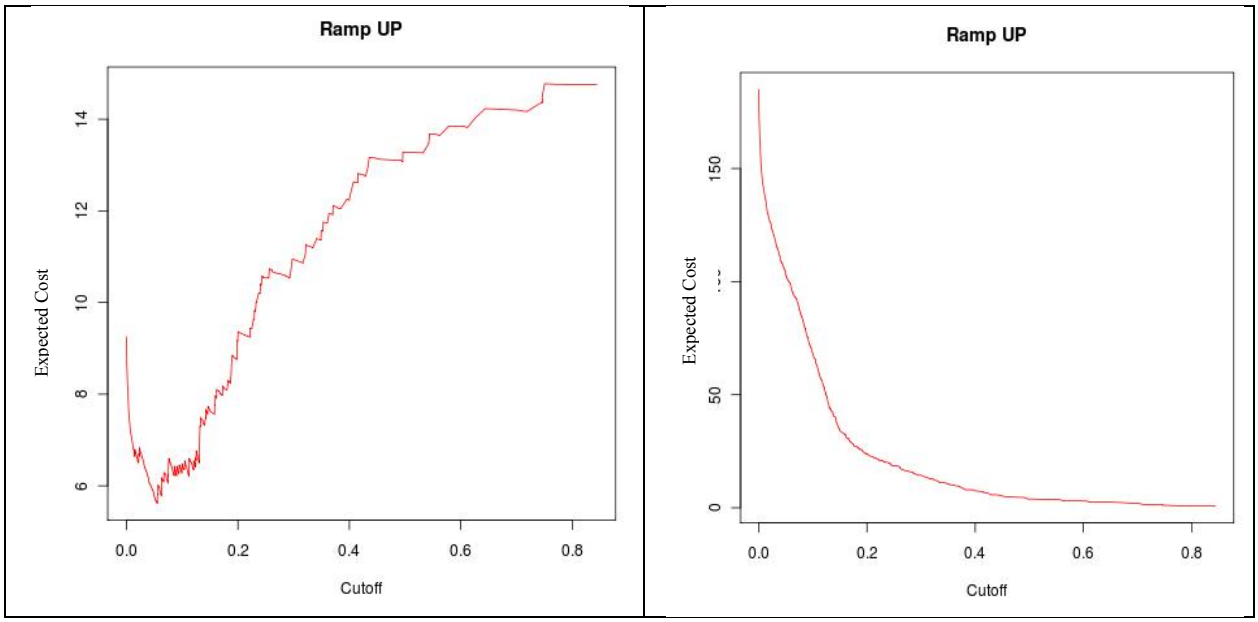

Fig. 4-27 Expected cost using definition 4: cFN=200 and cFP=10 (left), and cFN=10 and cFP=200 (right).

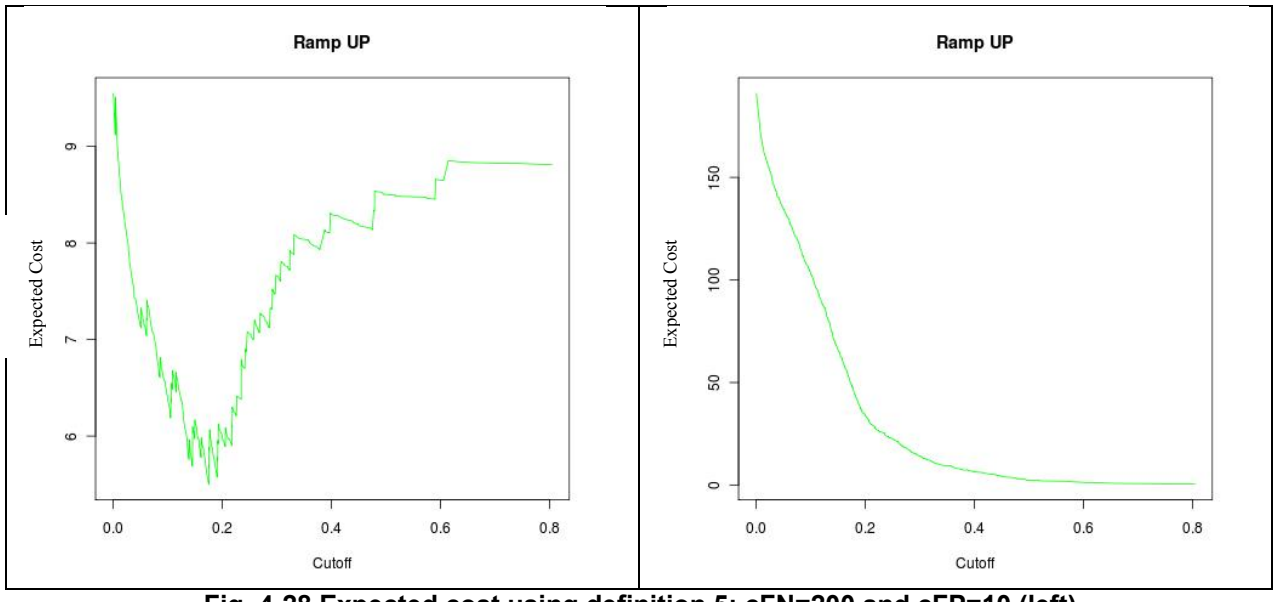

Fig. 4-28 Expected cost using definition 5: cFN=200 and cFP=10 (left), and cFN=10 and cFP=200 (right).

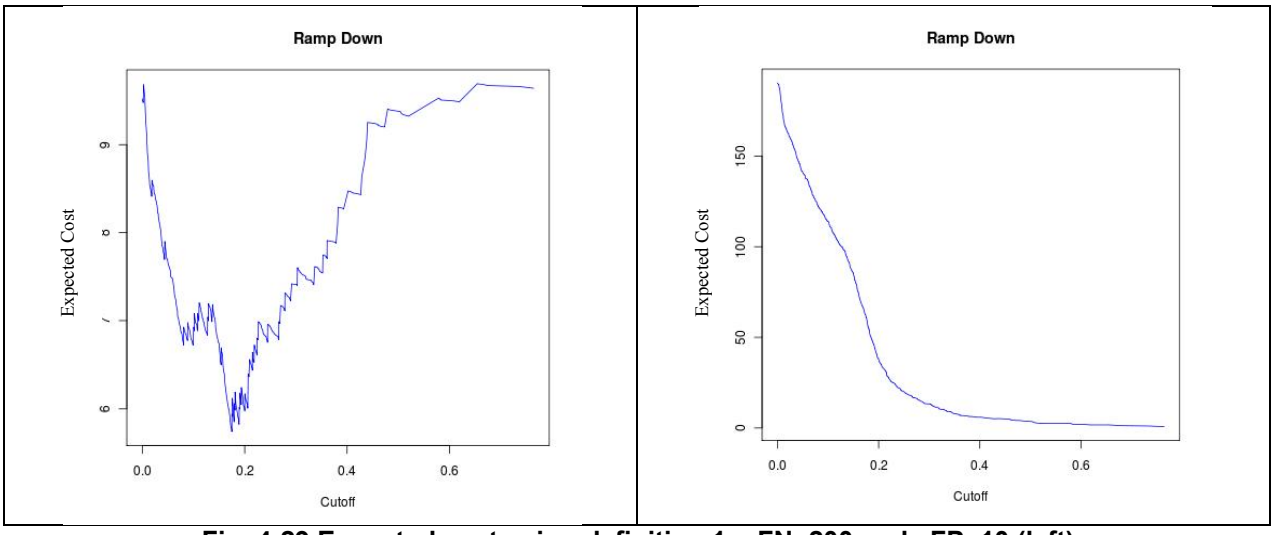

Fig. 4-29 Expected cost using definition 1: cFN=200 and cFP=10 (left), and cFN=10 and cFP=200 (right).

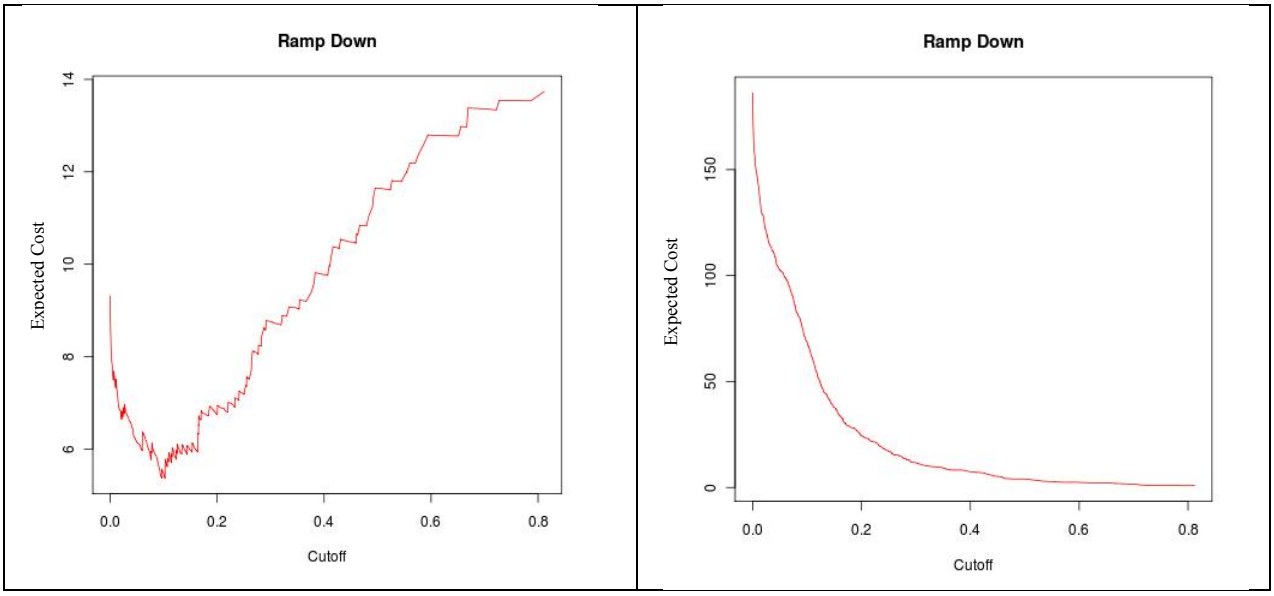

Fig. 4-30 Expected cost using definition 4: cFN=200 and cFP=10 (left), and cFN=10 and cFP=200 (right).

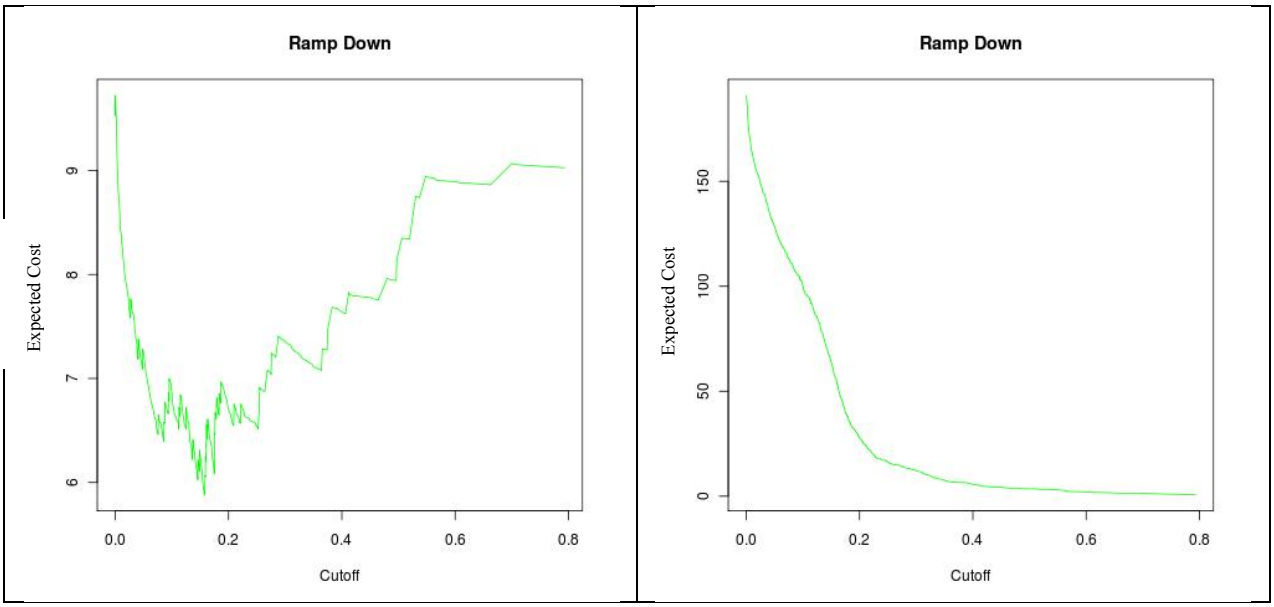

Fig. 4-31 Expected cost using definition 5: FN=200 and FP=10 (left), and FN=10 and FP=200 (right).

4.5.3.2 Analysis of the Results

In the development of our system, we use each one of the ramp definitions, a voting strategy, and a metric to find the optimum operating point. Regarding the definitions, we have three definitions that work over the power signal and two definitions that work over a filtered signal. The first three definitions are directly comparable, whereas the other two are more difficult to compare. One of the main issues is the size of the time step. In definitions 4 and 5, we do not have such parameters; however, in definition 4 we have the *nam* parameter, which defines the size of the interval and, therefore, the number of time points to include in the moving average. In definition 5, we have defined it by using one step ahead.

By setting the size of the step ahead to one, we obtain the same results when running definitions 1, 2, and 3. We obtain almost the same results when we run definition 4 with *nam=1*. In the case of definition 5, we almost always obtain better results than the ones obtained by running the other four definitions.

If we set the size of the step ahead to values higher than one, we obtain different results. The results of definition 2 are better than the ones obtained by definitions 1 and 3. The results obtained when running definition 3 are less comparable, as in this definition we are working with a ratio between power differences at the endpoints and the size of the time step ahead. If we change the power reference value accordingly, definition 1 is the result. Regarding definition 4, if we set *nam* parameter to values higher than one, we obtain better results than by using any of the other definitions.

Concerning the voting strategy that is at the core of our system and its success, we can say that by using it we can explore the diversity in the scenarios and generate valuable probabilistic forecasts. By using our voting strategy, in conjunction with a technique that we use to choose the optimum operating point, we realize significant improvements against the reference technique —

a point forecast system. For all metrics that we use to choose the optimum operating point, we obtain significant wins.

Despite the results, we test a set of metrics aiming to answer some open questions: What is the best metric that we can use to find the optimum operating point? How do we measure the performance of a probabilistic ramp forecasting system? Is there an overall best metric? Do any of the metrics' results reflect the aims and objectives of power system operators?

By inspecting the results of each metric, we obtain groups of metrics. In one group, we have CSI and F-Measure. By using each one of these two metrics, we find an optimum operating point (the probability threshold) which, when we run our system using that probability threshold, we obtain an almost equal number of forecast events and occurring events and a balanced number of FP and FN events. The second group of metrics includes EDS and KSS. When using these metrics to find the optimum operating point, we obtain a low-probability threshold. By using this operating point, we are always predicting a ramp event, a phenomena also known as hedging, and getting a large number of false positives. The third group includes the OR metric. By using this metric, two cases result: (1) if we do not consider phase errors, we obtain a large operating point and, therefore, a small number of events and FPs; and (2) if we consider phase errors, an OR curve having a shape similar to a parabola results. In the fourth group, we have the ROC curves. By using these curves, we can access the performance of the ramp forecast system under varying costs of predicting the wrong class. This way, we can include economic information in the search for the optimum operating point. In conjunction with the expected cost formula, we obtain a decision-making framework that is flexible and can include valuable economic knowledge. If operated by an experienced technician who knows the costs of making prediction errors, by inspecting the expected cost graph, the operator can both minimize the expected cost and analyze the risk of each decision.

Thus, by presenting this report, we expect that experienced operators can obtain help in deciding which metric best fits their operations. In our opinion, having the ability to inspect the range of FPs and FNs that can be obtained by using any of the metrics of the first three groups points to the development and use of a flexible system, such as the ROC curves and expected cost curves. By using these techniques, we add flexibility to our system that will be difficult to surpass.

Regarding the study of the other parameters, we can say that we experience better predictions with both types of ramps if we use large sizes of the time step. This result is particularly clear when predicting ramp-down events. This phenomenon can be explained by the smooth changes in the wind, especially in downward ramps. Concerning the aggregation, we also obtain better results if we work with large aggregation intervals.

As expected, if we analyze the results obtained by allowing the phase error, we realize important gains in performance metrics. If time precision can be relaxed, then the phase error technique should be used.

In summary, we highlight the main findings:

- The probabilistic ramp event forecast system presented in this chapter proves that it can generate better ramp predictions than a point forecast system over a range of metrics. Moreover, by using a probabilistic ramp event forecasting system, the user can perform a risk analysis.
- The vertical histograms give additional valuable information. For instance, if a wide number of scenarios predict a large ramp, the probability of correctly predicting a ramp is higher.
- The ROC curves and Expected Cost framework have the flexibility to assess the performance of forecasters and allow us to find the optimum operating point. Their use is intuitive and can provide business information.
- If the prediction timing of each event can be relaxed, the phase error technique must be used. By using this technique, we can obtain important gains.
- There are several ramp event definitions, and the best ramp definition to use at a specific wind farm depends on what the forecast will be used for and can be dependent on the terrain conditions, equipment technical requirements, and other operation conditions. Despite this observation, definitions 4 and 5 seem to provide better results according to the majority of the metrics tested.
- The downward ramps are easier to predict than the upward ones. This result is especially clear if we consider large sizes of the time step.

Overall, we can say that we achieve better results by using large sizes of the aggregation window and large sizes of the time step.

4.6 Conclusions

Ramp event forecasting is an important topic, and the number of works addressing this problem is rapidly increasing. In this chapter, we presented a new definition of ramp events based on high-pass filters — a new method for probabilistic ramp event detection based on scenarios — and we performed an extensive experimental evaluation using some recently proposed ramp definitions. Another contribution of this work is the visualization method. Probabilistic ramp events are presented using histograms of cumulative ramp probability functions. Each bin corresponds to ramp magnitude, and the corresponding value is the probability of observing a ramp event of magnitude equal to or greater than that of magnitude. We should stress that the proposed method is independent of a particular ramp event definition and can be implemented using any ramp definition.

The analysis of the performance of the method indicates that we obtained important gains when we compare our system against a reference model (i.e., a point forecast system). So far, the success of ramp prediction methods reported in the literature is not impressive, and some failure may be explained by phase errors inherited from the meteorological wind prediction models. A study on this factor was also conducted in this chapter. The study shows that by using a technique to correct phase errors, we can realize important gains.

In the last phase of the development of this work, we identified a new and promising methodology that learns weights for each bin in the vertical histogram and then uses a weighted voting to generate probabilities. We have already developed a prototype that uses this methodology and have run some experiments. In a preliminary analysis, we can say that we obtain some improvements over the model described in this chapter. Therefore, this study delivers advances to the state-of-the-art and provides strong directions for future work.

5 GENERAL CONCLUSIONS

In this report, we have documented our work on developing improved statistical methods for WPF. The main conclusions and contributions to the current state-of-the-art for each area of research are briefly summarized below.

For *wind power point forecasting*, we focused on the training criteria used in the computational learning algorithms (we used a neural network), which converts weather forecasts and observational data to a point forecast for wind power. In particular, we focused on the use of ITL training criteria, which are not built on the assumption of a Gaussian distribution of the forecasting errors. Through extensive testing of different training criteria on a large-scale wind farm located in the U.S. Midwest, we found that:

- The ITL training criteria showed a favorable performance in terms of lower forecasting errors compared to the classical MSE criterion;
- The improvements of the ITL criteria were particularly significant for low and high wind power output levels;
- A new ITL-based training criterion, centered correntropy, was introduced for the first time in this report;
- Among several ITL-based criteria, the maximum correntropy criterion (MCC) showed good results and also has a low computational burden;
- The results showed evidence that online training assures better results in the presence of concept drift in the training data.

Within *wind power uncertainty forecasting*, we developed and tested novel time-adaptive algorithms based on KDF. We developed a time-adaptive KDF model using the NW estimator. Furthermore, we applied the QC estimator for the first time to the WPF problem. In both cases, we paid particular attention to the choice of adequate kernels. Through extensive testing and comparison to QR on several datasets, we found that:

- KDF methods showed a tendency to present a better performance than QR in terms of calibration;
- The QR methods showed a tendency to present a better performance in terms of sharpness and resolution;
- The aggregate skill score of the QR and KDF methods were rather similar;
- The application of the time-adaptive KDF models improved the results compared to offline training of the KDF algorithms;
- Adequate kernel selection for each variable proved to be very important in KDF, both in terms of kernel type and kernel size;
- An important advantage of KDF is that it estimates the full probability distribution for wind power at any forecast horizon.

For *wind power ramp forecasting*, we presented a new definition of ramp events based on high-pass filters, a new method for probabilistic ramp event detection based on scenarios, and performed an extensive experimental evaluation using some recently proposed ramp definitions. From the analysis and results, we conclude that:

- The proposed probabilistic method obtained important gains when compared to a reference model (i.e., a point forecast system);
- By using a technique to correct phase errors, we obtrained important gains in forecasting performance;
- A number of different ramp definitions exist. The proposed method is independent of a particular ramp event definition and can be implemented using any definition;
- The visualization of the ramp forecast is important. In the proposed method, probabilistic ramp events are presented using histograms of cumulative ramp probability functions.

Finally, most of the WPF prototypes and algorithms developed in the project and used to generate the results documented in this report have been integrated into a software platform for WPF research named "ARGUS-PRIMA." More information about the platform can be obtained by contacting Argonne National Laboratory.

6 REFERENCES

[1] C. Monteiro, R. Bessa, V. Miranda, A. Botterud, J. Wang, G. Conzelmann, "Wind Power Forecasting: State-of-the-Art 2009," Argonne National Laboratory, Report ANL/DIS-10-1, 2009. Available at: http://www.dis.anl.gov/projects/windpowerforecasting.html.

[2] A. Botterud, Z. Zhou, J. Wang, R.J. Bessa, J. Mendes, J. Sumaili, V. Miranda, "Use of Wind Power Forecasting in Operational Decisions," Technical Report, Argonne National Laboratory and INESC Porto, Sept. 2011.

[3] Official website of PostgreSQL, open-source database. Available at: http://www.postgresql.org/.

[4] Official website of WRF – Weather Research and Forecasting Model, a mesoscale numerical weather prediction system. Available at: http://www.wrf-model.org/.

[5] S. Haykin, *Neural Networks and Learning Machines* – 3rd Edition, Pearson Education, New Jersey, 2009.

[6] C. Igel, M. Hüsken, "Improving the Rprop learning algorithm," In H. Bothe and R. Rojas, editors, *Proceedings of the Second International ICSC Symposium on Neural Computation (NC 2000)*, pp. 115–121, ICSC Academic Press, 2000.

[7] Boost C++ Libraries. Available at: http://www.boost.org/.

[8] J.C. Principe, D. Xu, "Information-theoretic learning using Renyi's quadratic entropy," in J.-F. Cardoso, C. Jutten, and P. Loubaton, editors, *Proceedings of the First International Workshop on Independent Component Analysis and Signal Separation, Aussois*, France, pages 407–412, 1999.

[9] J.M. Santos, J. Marques de Sá, L.A. Alexandre, "Batch-sequential algorithm for neural networks trained with entropic criteria," *Proceedings of the 15th Int. Conf. on Artificial Neural Networks*, (p. 91–96), 2005.

[10] R.J. Bessa, V. Miranda, J.C. Principe, A. Botterud, J. Wang, "Information Theoretic Learning applied to Wind Power Modeling," *2010 IEEE World Congress on Computational Intelligence*, Barcelona, Spain, Jul. 2010.

[11] R.J. Bessa, V. Miranda, A. Botterud, J. Wang, "'Good' or 'Bad' Wind Power Forecasts: A Relative Concept," *Wind Energy*, Vol. 14, No. 5, pp. 625–636, 2011.

[12] G. Welch, G. Bishop, "An Introduction to the Kalman Filter," *SIGGRAPH*, 2001. Available at: http://www.cs.unc.edu/~welch/kalman/.

[13] D. Simon, "Kalman Filtering," *Embedded Systems Programming*, 2001. Available at: http://calypso.inesc-id.pt/FCUL/psm/docs/kalman-dan-simon.pdf.

[14] J. Juban, L. Fugon, G. Kariniotakis, "Uncertainty estimation of wind power forecasts," in *Proceedings of the European Wind Energy Conference EWEC'08*, Brussels, Belgium, March 31–April 03, 2008.

[15] P. Pinson, *Estimation of the uncertainty in wind power forecasting*, Ph.D. dissertation, Ecole des Mines de Paris, 2006.

[16] J.B. Bremnes, "Probabilistic wind power forecasts using local quantile regression," *Wind Energy*, vol. 7, no. 1, pp. 47–54, 2004.

[17] J. Juban, N. Siebert, G. Kariniotakis, "Probabilistic short-term wind power forecasting for the optimal management of wind generation," in *Proceedings of the IEEE Power Tech Conference*, Lausanne, Switzerland, July 2007.

[18] J.K. Møller, H.A. Nielsen, H. Madsen, "Time-adaptive quantile regression," *Computational Statistics & Data Analysis*, vol. 52, no. 3, pp. 1292–1303, Jan. 2008.

[19] P. Pinson, "On probabilistic forecasting of wind power time-series," submitted to *Wind Energy*, April 26 2010.

[20] J.B. Bremnes, "Probabilistic wind power forecasts using local quantile regression," *Wind Energy*, vol. 7, pp. 47–54, 2004.

[21] A. Botterud, J. Wang, R. Bessa, H. Keko, V. Miranda, "Risk management and optimal bidding for a wind power producer," in *Proceedings of the IEEE PES General Meeting*, Minneapolis, USA, 2010.

[22] F. Bourry, J. Juban, L.M. Costa, G. Kariniotakis, "Advanced strategies for wind power trading in short-term electricity markets," in *Proceeding of the European Wind Energy Conference & Exhibition EWEC 08*, Brussels, Belgium, March 31– April 3, 2008.

[23] M.A. Matos, "Decision under risk as a multicriteria problem," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1516–1529, Sept. 2007.

[24] M.A. Matos, R. Bessa, "Setting the operating reserve using probabilistic wind power forecasts," *IEEE Transactions on Power Systems*, vol. 26, no. 2, pp. 594-603, 2011.

[25] R. Bessa, M.A. Matos, "Comparison of probabilistic and deterministic approaches for setting operating reserve in systems with high penetration of wind power," in *Proceedings of the 7th Mediterranean Conference and Exhibition on Power Generation, Transmission, Distribution and Energy Conversion — MedPower2010*, Agia Napa, Cyprus, Nov. 7–10, 2010.

[26] P. Pinson, G. Papaefthymiou, B. Klockl, H.Aa. Nielsen, H. Madsen, "From probabilistic forecasts to statistical scenarios of short-term wind power production," *Wind Energy*, vol. 12, no. 1, pp. 51–62, 2009.

[27] H.A. Nielsen, H. Madsen, T.S. Nielsen, "Using quantile regression to extend an existing wind Power Forecasting system with probabilistic forecasts," *Wind Energy*, vol. 9, no. 1–2, pp. 95–108, 2006.

[28] D.M. Bashtannyk, R.J. Hyndman, "Bandwidth selection for kernel conditional density estimation," *Computational Statistics & Data Analysis*, vol. 36, pp. 279–298, 2001.

[29] J.G. Gooijer, D. Zerom, "On conditional density estimation," *Statistica Neerlandica*, vol. 57, pp. 159–176, 2003.

[30] O.P. Faugeras, "Prediction via the quantile-copula conditional density estimator," *Toulouse School of Economics Working Papers Series*, no. 09-124, Dec. 2009.

[31] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.

[32] E. Parzen, "On Estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[33] C. Wolverton, T.J. Wagner, "Asymptotically optimal discriminant functions for pattern classification," *IEEE Transactions on Information Theory*, vol. 15, no. 2, pp. 258–265, March 1969.

[34] M.P. Wand, M.C. Jones, "Multivariate plug-in bandwidth selection," *Computational Statistics*, vol. 9, pp. 97–116, 1994.

[35] M. Rosenblatt, *Conditional probability density and regression estimators*, in Multivariate Analysis II, New York: Academic Press, pp. 25–31, 1969.

[36] R.J. Hyndman, D.M. Bashtannyk, G.K. Grunwald, "Estimating and visualizing conditional densities," *Journal of Computational and Graphical Statistics*, vol. 5, no. 4, pp. 315–336, Dec. 1996.

[37] O.P. Faugeras, "A quantile-copula approach to conditional density estimation," *Journal of Multivariate Analysis*, vol. 100, pp. 2083–2099, Oct. 2009.

[38] T. Bouezmarni, J.V.K. Rombouts, "Semiparametric multivariate density estimation for positive data using copulas," *Computational Statistics & Data Analysis*, vol. 53, no. 6, pp. 2040–2054, Apr. 2009.

[39] M. Sklar, "Fonctions de répartition à n dimensions et leurs marges," *Publications de l'Institut de Statistique de L' Université de Paris*, vol. 8, pp. 229–231, 1959.

[40] R. Nelson, *An Introduction to Copulas. Lecture Notes in Statistics*, New York: Springer, vol. 139, 1999.

[41] G. Frahm, M. Junker, A. Szimayer, "Elliptical copulas: applicability and limitations," *Statistics & Probability Letters*, vol. 63, no. 3, pp. 275–286, July 2003.

[42] S.X. Chen, "Beta kernel estimators for density functions," *Computational Statistics & Data Analysis,* vol. 31, no. 2, pp. 131–145, 1999.

[43] S. Zhang, R.J. Karunamuni, "On kernel density estimation near endpoints," *Journal of Statistical Planning and Inference*, vol. 70, no. 2, pp. 301–316, Jul. 1998.

[44] M.C. Jones, D.A. Henderson, "Miscellanea kernel-type density estimation on the unit interval," *Biometrika*, *Oxford University Press for Biometrika Trust*, vol. 94, no. 4, pp. 977–984, 2007.

[45] S. Zhang, "Boundary performance of the beta kernel estimators," *Journal of Nonparametric Statistics*, vol. 22, no. 1, pp. 81–104, Jan. 2010.

[46] C. Gourieroux, A. Monfort, "(Non) consistency of the beta kernel estimator for recovery rate distribution," Working Paper Nº2006-31, Istitut National de la Statistique et des Etudes Economiques, Dec. 2006.

[47] S.X. Chen, "Probability density function estimation using gamma kernels," *Annals of the Institute of Statistical Mathematics*, vol. 52, no. 3, pp. 471–480, Sept. 2000.

[48] S. Zhang, "A note on the performance of the gamma kernel estimators at the Boundary," *Statistics and Probability Letters*, vol. 80, no. 7–8, pp. 548–557, April 2010.

[49] K.V. Mardia, P.E. Jupp, *Directional statistics*, Wiley's Series in Probability and Statistics, Nov. 1999.

[50] E.J. Wegman, H.I. Davies, "Remarks on recursive estimators of a probability density," *The Annals of Statistics*, vol. 7, no. 2, pp. 316–327, 1979.

[51] E.J. Wegman, D.J. Marchette, "On some techniques for streaming data: a case study of Internet packet headers," *Journal of Computational and Graphical Statistics*, vol. 12, no. 4, pp. 893–914, 2003.

[52] K.A. Caudle, E.J. Wegman, "Nonparametric density estimation of streaming data using orthogonal series," *Computational Statistics and Data Analysis*, vol. 53, pp. 3980–3986, 2009.

[53] P. Pinson, G. Kariniotakis, H.A. Nielsen, T.S. Nielsen, H. Madsen, "Properties of quantile and interval forecasts of wind generation and their evaluation," in *Proceedings of the European Wind Energy Conference EWEC'06*, Athens, Greece, 2006.

[54] P. Pinson, H.A. Nielsen, J.K. Moller, H. Madsen, G. Kariniotakis, "Nonparametric probabilistic forecasts of wind power: required properties and evaluation," *Wind Energy*, vol. 10, no. 6, pp. 497–516, Nov. 2007.

[55] T. Gneiting, A.E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *Journal of the American Statistical Association*, vol. 102, no. 477, pp. 359–378, March 2007.

[56] Eastern Wind Integration and Transmission Study (EWITS), National Renewable Energy Laboratory (NREL). Available at: http://www.nrel.gov/wind/systemsintegration/ewits.html.

[57] M. Brower, "Development of Eastern Regional Wind Resource and Wind Plant Output Datasets," NREL Subcontract Report NREL/SR-550-46764, December 2009.

[58] R.J. Hyndman, J. Einbeck, M. Wand, "Package 'hdrcde,'" *R Manual*, April 2010. Available at: http://www.robjhyndman.com/software/hdrcde.

[59] V.R. Jose, R.L. Winkler, "Evaluating quantile assessments," *Operations Research*, vol. 57, no. 5, pp. 1287–1297, Sept. 2009.

[60] D. Maggio, "Integrating wind forecasting into market operation - ERCOT," presentation at Utility Wind Integration Group (UWIG) Workshop, Phoenix, AZ, United States, Feb. 18–19, 2009.

[61] J. Parkes, J. Wasey, A. Tindal, "Wind Energy Trading Benefits through Short-Term Forecasting," in *European Wind Energy Conference*, 2006.

[62] C. Kamath, "Understanding Wind Ramp Events Through Analysis of Historical Data," in *IEEE PES Transmission and Distribution Conference and Expo*, New Orleans, LA, United States, 2010.

[63] N. Francis, "Predicting Sudden Changes in Wind Power Generation," in *North American WindPower*, 2008.

[64] B. Greaves, J. Collins, J. Parkes, A. Tindal, "Temporal Forecast Uncertainty for Ramp Events," in *European Wind Energy Conference 2009 (EWEC)*, Marseille, France, 2009.

[65] U. Focken, M. Lange, "Wind power forecasting pilot project in Alberta," Oldenburg, Germany: energy & meteo systems GmbH, 2008.

[66] A. Kusiak, H. Zheng, "Prediction of Wind Farm Power Ramp Rates: A Data-Mining Approach," *Journal of Solar Energy Engineering*, vol. 131, 2009.

[67] AWS Truewind-LLC, "AWS Truewind's Final Report for the Alberta Forecasting Pilot Project," Alberta, Canada, 2008.

[68] N. Cutler, M. Kay, H. Outhred, I. MacGill, "High-Risk Scenarios for Wind Power Forecasting in Australia," in *Proceedings of the European Wind Energy Conference & Exhibition*, 2007.

[69] N. Cutler, M. Kay, K. Jacka, T.S. Nielsen, "Detecting, Categorizing, and Forecasting Large Ramps in Wind Farm Power Output Using Meteorological Observations and WPPT," *Wind Energy*, pp. 453–470, 2007.

[70] J. Parkes, J. European Wind Energy Conference, March 2009. Available at: http://www.ewec2009proceedings.info/allfiles/205_EWEC2009presentation.ppt.

[71] C.W. Potter, E. Grimit, B. Nijssen, "Potential Benefits of a Dedicated Probabilistic Rapid Ramp Event Forecast Tool," *IEEE*, 2009.

[72] J. Freedman, M. Markus, R. Penc, "Analysis of West Texas Wind Plant Ramp-up and Ramp-down Events," AWS Truewind, LLC, Albany, NY, United States, 2008.

[73] J.W. Zack, S. Young, M. Cote, J. Nocera, "Development and Testing of an Innovative Short-Term Large Wind Ramp Forecasting System," in *Wind Power Conference & Exhibition*, Dallas, Texas, United States, 2010.

[74] A. Bossavy, R. Girard, G. Kariniotakis, "Forecasting Uncertainty Related to Ramps of Wind Power Production," in *Proceedings of the European Wind Energy Conference*, Warsaw, Poland, 2010.

[75] G. Kariniotakis, I. Martí, D. Casas, P. Pinson, T.S. Nielsen, H. Madsen, G. Giebel, J. Usaola, I. Sanchez, A.M. Palomares, R. Brownsword, J. Tambke, U. Focken, M. Lange, P. Louka, G. Kallos, C. Lac, G. Sideratos, G. Descombes, "What performance can be expected by short-term wind power prediction models depending on site characteristics?," in *European Wind Energy Conference*, London, UK, 2004.

[76] P.Pinson, H. Madsen, H. A. Nielsen, Papaefthymiou, B. Klockl, "From Probabilistic Forecasts to Statistical Scenarios of Short-term Wind Power Production," *Wind Energy*, vol. 12, pp. 51–62, 2009.

[77] TESLA, Inc. Available at: http://www.teslaforecast.com/TeslaModel.aspx.

[78] M.G. De Giorgi, A. Ficarella, M. Tarantino, "Error analysis of short-term wind power prediction models," *Applied Energy*, vol. 88, no. 4, pp. 1298–1311, 2011.

[79] P. Pinson, "Estimation of the uncertainty in wind power forecasting," École des Mines de Paris, Paris, PhD Dissertation, 2006.

[80] D. Hand, "Measuring classifier performance: a coherent alternative to the area under the ROC curve," *Machine Learning*, vol. 77, pp. 103–123, 2009.

[81] F. Provost, T. Fawcett, "Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions," in *Third International Conference on Knowledge Discovery and Data Mining (KDD)*, California, USA, 1997, pp. 43–48.

[82] C.J. Rijsbergen, *Information Retrieval*. London: Butterworths, 1979.

[83] J.T. Schaefer, "The Critical Success Index as an Indicator of Warning Skill," *Weather Forecasting*, no. 5, pp. 570–575, 1990.

[84] K.T. Bradford, R.L. Carpenter, B. Shaw, “Forecasting Southern Plains Wind Ramp Events Using the WRF Model at 3-km,” in *AMS Student Conference*, 2010.

[85] A.W. Hanssen, W.J.A. Kuipers, “On the relationship between the frequency of rain and various meteorological parameters,” *Mededelingen van de Verhandlungen*, vol. 81, pp. 2–15, 1965.

[86] A. Ghelli, C. Primo, “On the use of the extreme dependency score to investigate the performance of an NWP model for rare events,” *Meteorological Applications*, vol. 16, pp. 537–544, 2009.

[87] G. Brier, “Verification of forecasts expressed in terms of probability,” *Monthly weather review*, vol. 78, no. 1, pp. 1–3, 1950.

This page intentionally blank

APPENDIX A – EVALUATION RESULTS FOR NREL DATASET OFFLINE TESTS

Figs. A-1 through A-10 present results using data from the National Renewable Energy Laboratory (NREL).

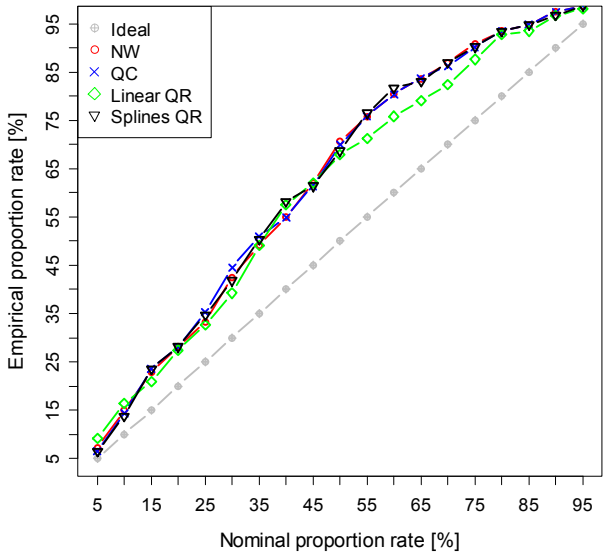

Fig. A-1 Calibration diagram for look-ahead time step t+6h (NREL data).

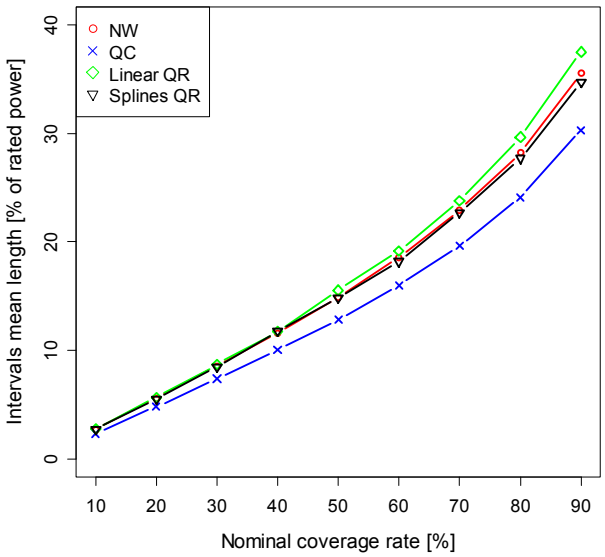

Fig. A-2 Sharpness diagram for look-ahead time step t+6h (NREL data).

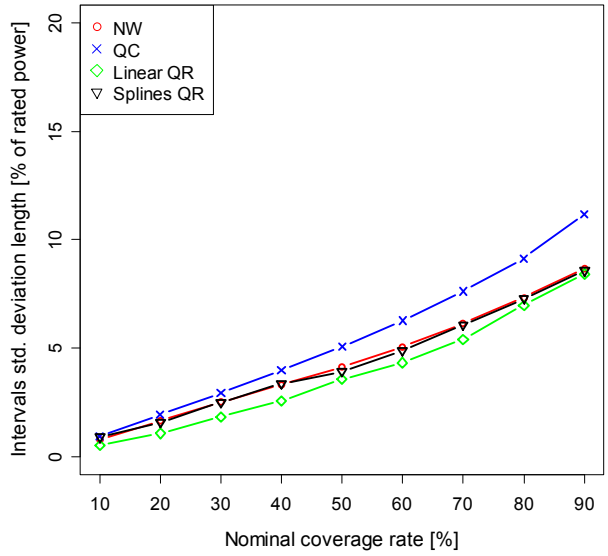

Fig. A-3 Resolution diagram for look-ahead time step t+6h (NREL data).

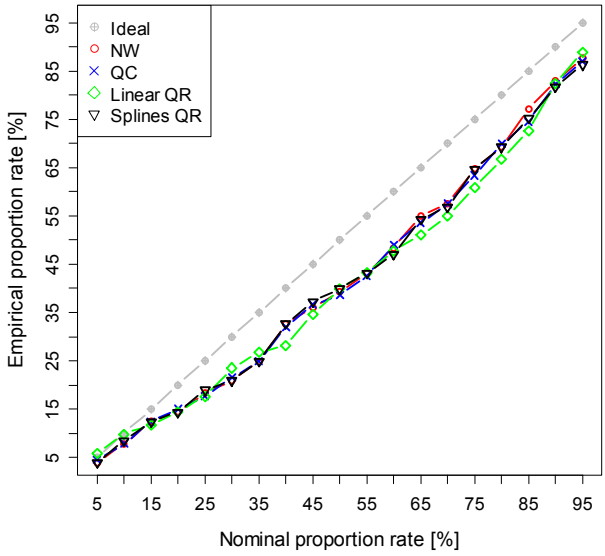

Fig. A-4 Calibration diagram for look-ahead time step t+22h (NREL data).

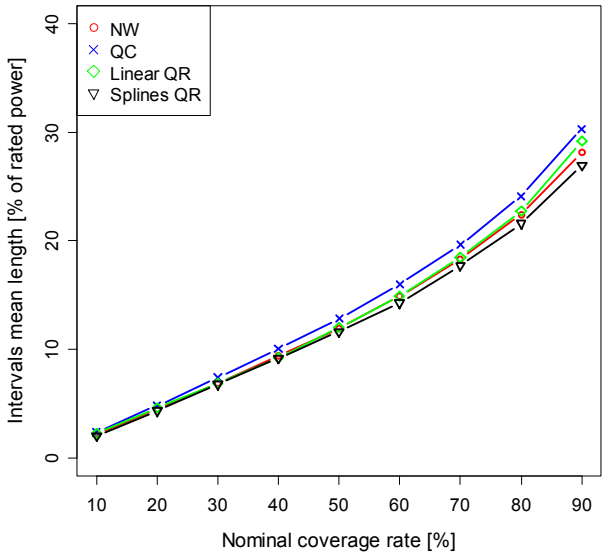

Fig. A-5 Sharpness diagram for look-ahead time step t+22h (NREL data).

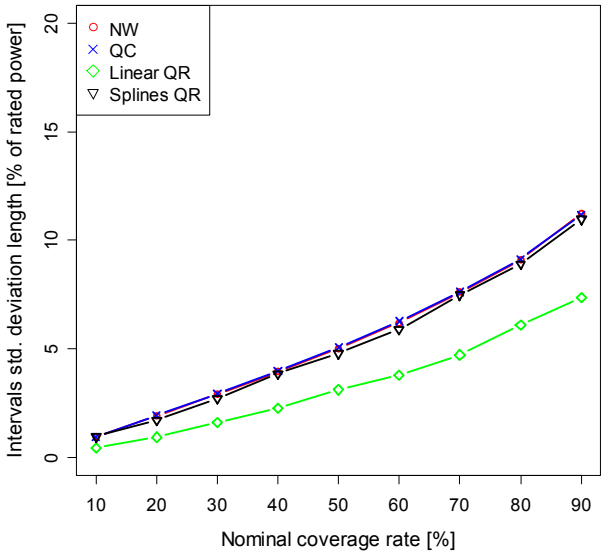

Fig. A-6 Resolution diagram for look-ahead time step t+22h (NREL data).

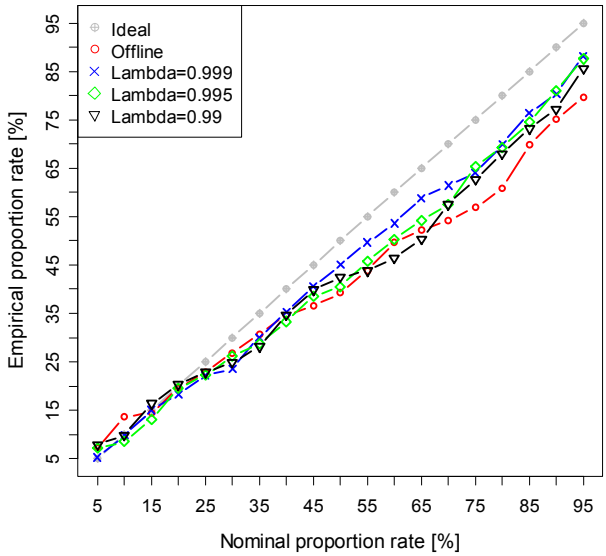

Fig. A-7 Calibration diagram for the NREL dataset with concept change and NW estimator for look-ahead time step *t+15h*.

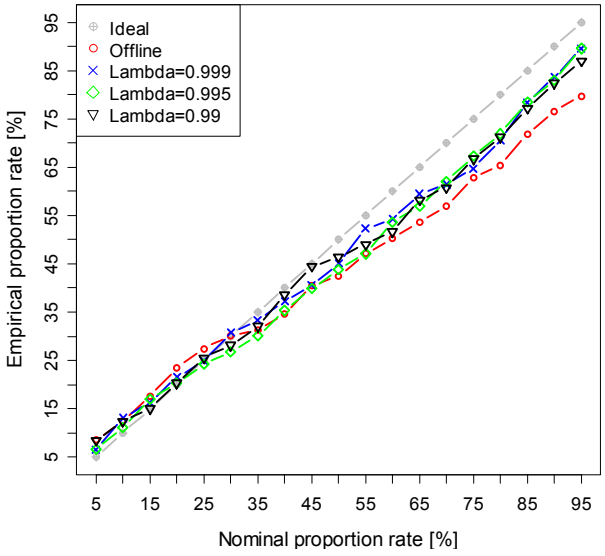

Fig. A-8 Calibration diagram for the NREL dataset with concept change and NW estimator for look-ahead time step t+20h.

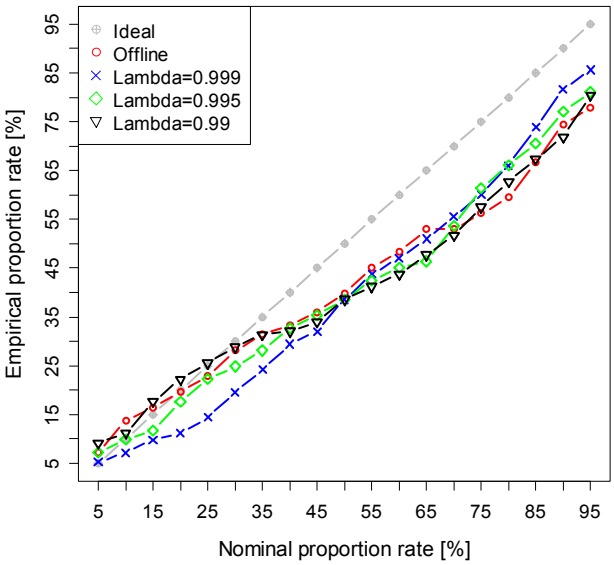

Fig. A-9 Calibration diagram for the NREL dataset with concept change and QC estimator for look-ahead time step t+15h.

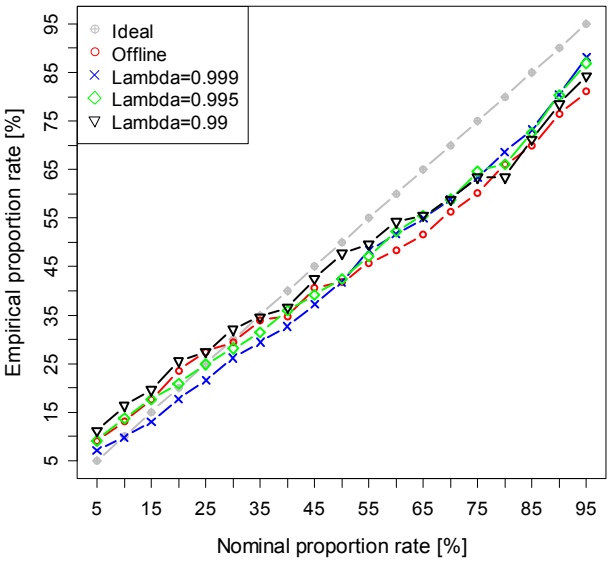

Fig. A-10 Calibration diagram for the NREL dataset with concept change and QC estimator for look-ahead time step t+20h.

This page intentionally blank

APPENDIX B – OFFLINE EVALUATION RESULTS FOR WIND FARM A

Figs. B-1 through B-12 present offline results for Wind Farm A.

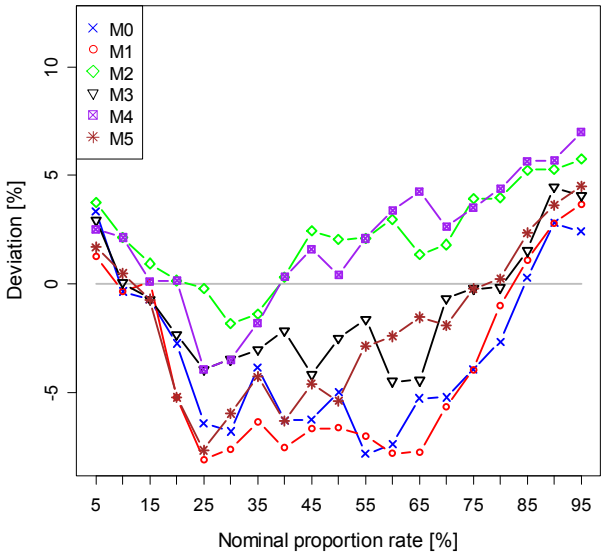

Fig. B-1 Calibration diagram for WFA with 6:00 AM NWP and NW models M0–M5 for look-ahead time step t+15h.

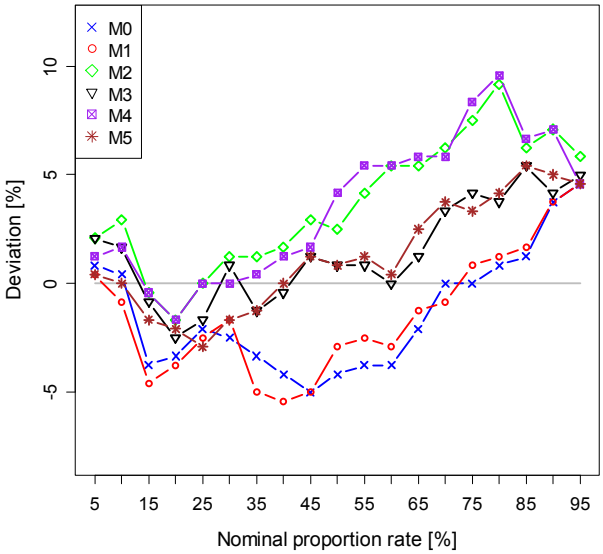

Fig. B-2 Calibration diagram for WFA with 6:00 PM NWP and NW models M0–M5 for look-ahead time step t+15h.

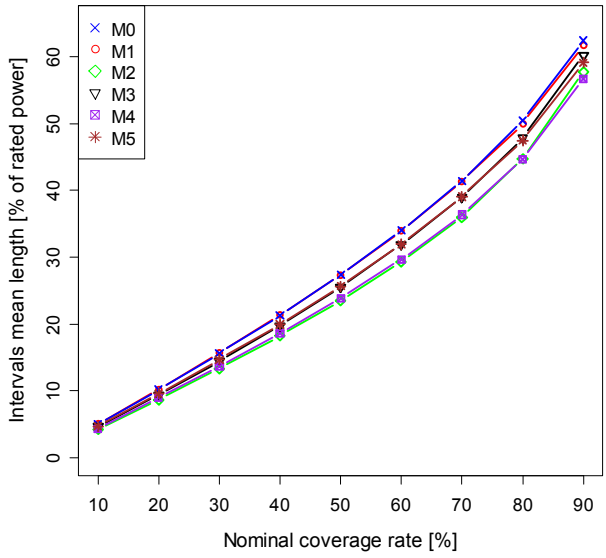

Fig. B-3 Sharpness diagram for WFA with 6:00 AM NWP and NW models M0–M5 for look-ahead time step t+15h.

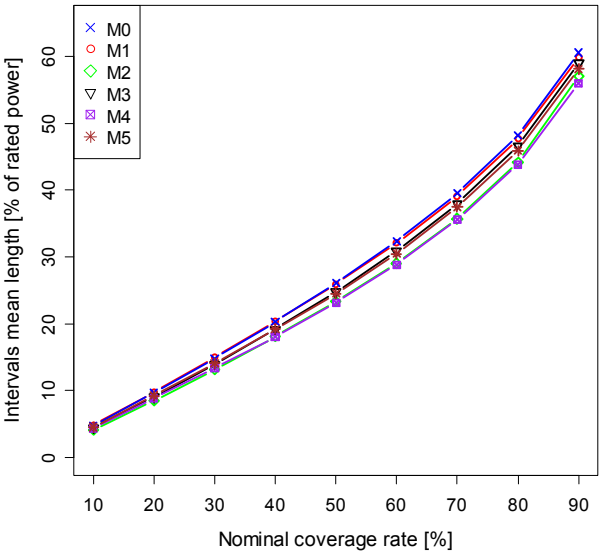

Fig. B-4 Sharpness diagram for WFA with 6:00 PM NWP and NW models M0–M5 for look-ahead time step t+15h.

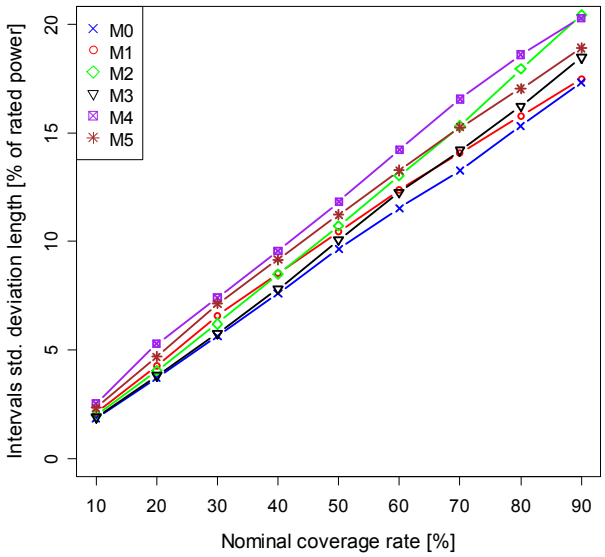

Fig. B-5 Resolution diagram for WFA with 6:00 AM NWP and NW models M0–M5 for look-ahead time step t+15h.

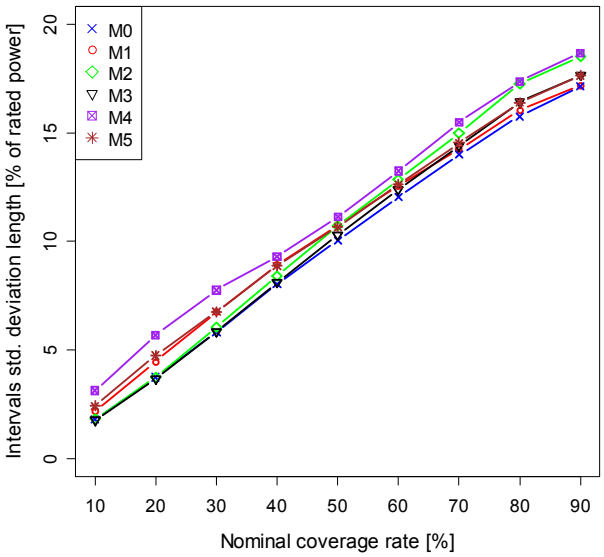

Fig. B-6 Resolution diagram for WFA with 6:00 PM NWP and NW models M0–M5 for look-ahead time step t+15h.

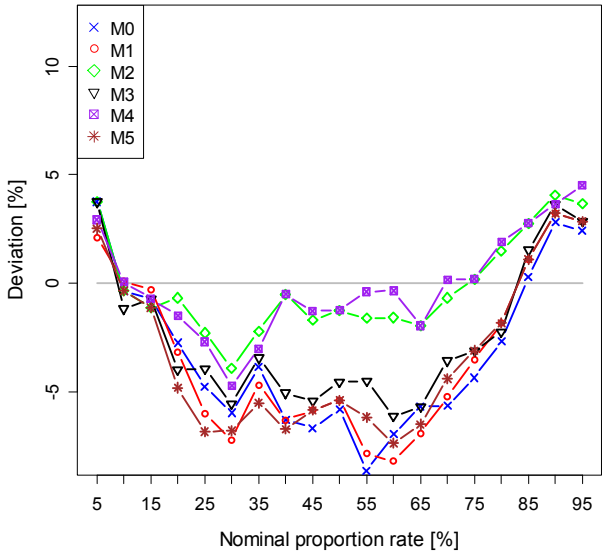

Fig. B-7 Calibration diagram for WFA with 6:00 AM NWP and QC models M0–M5 for look-ahead time step t+15h.

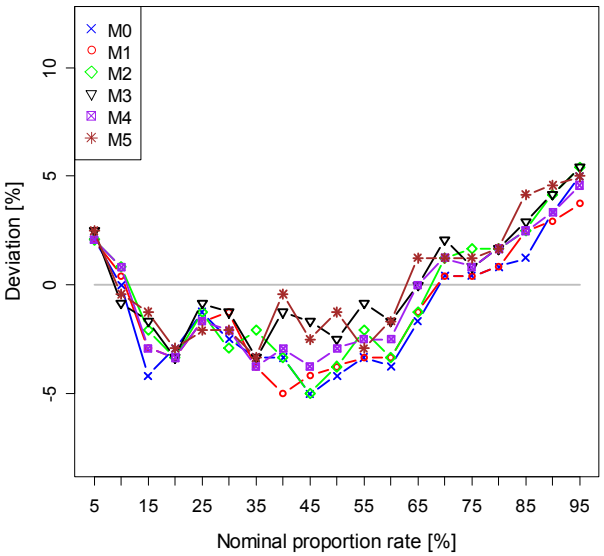

Fig. B-8 Calibration diagram for WFA with 6:00 PM NWP and QC models M0–M5 for look-ahead time step t+15h.

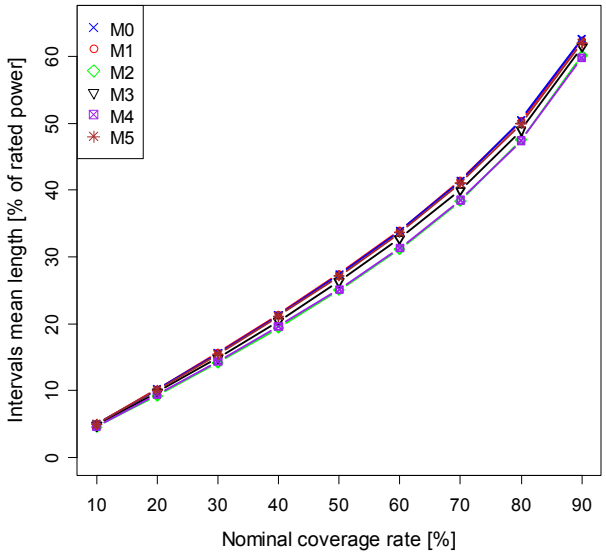

Fig. B-9 Sharpness diagram for WFA with 6:00 AM NWP and QC models M0–M5 for look-ahead time step t+15h.

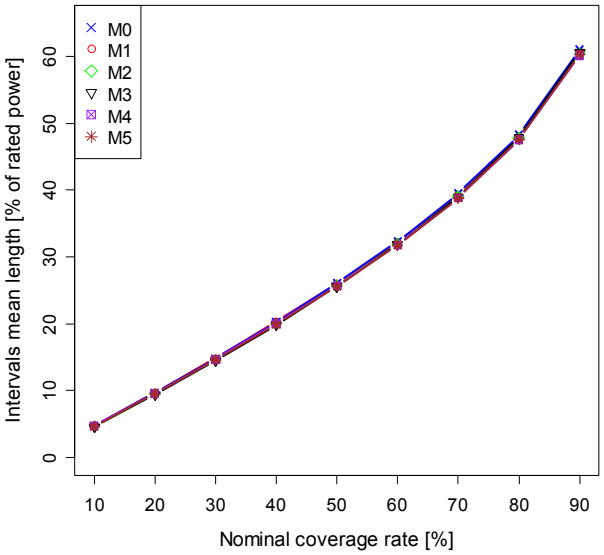

Fig. B-10 Sharpness diagram for WFA with 6:00 PM NWP and QC models M0–M5 for look-ahead time step t+15h.

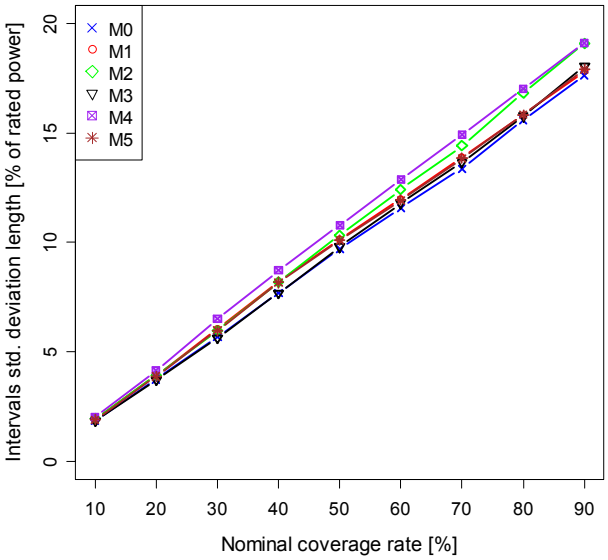

Fig. B-11 Resolution diagram for WFA with 6:00 AM NWP and QC models M0–M5 for look-ahead time step t+15h.

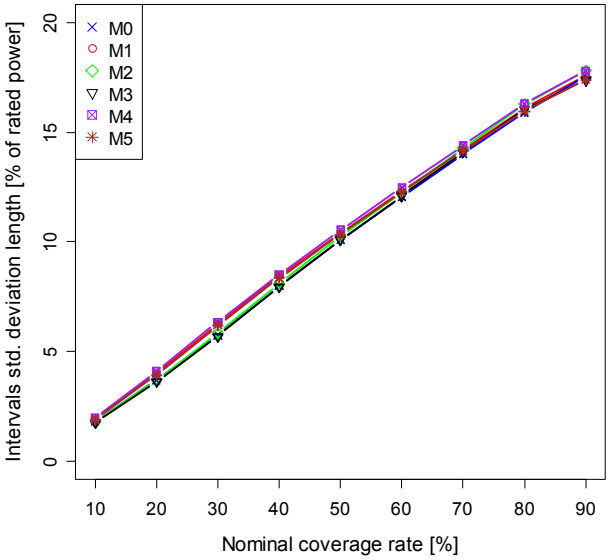

Fig. B-12 Resolution diagram for WFA with 6:00 PM NWP and QC models M0–M5 for look-ahead time step t+15h.

This page intentionally blank

APPENDIX C – OFFLINE EVALUATION RESULTS FOR WIND FARM B

Figs. C-1 through C-12 present offline results for Wind Farm B.

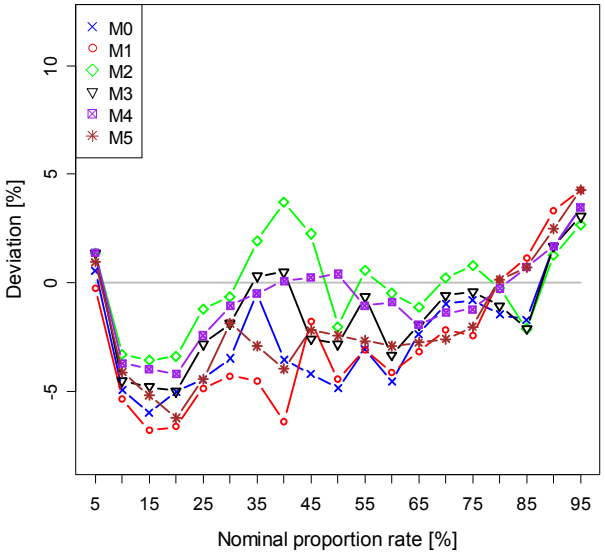

Fig. C-1 Calibration diagram for WFB with 6:00 AM NWP and NW models M0–M5 for look-ahead time step t+15h.

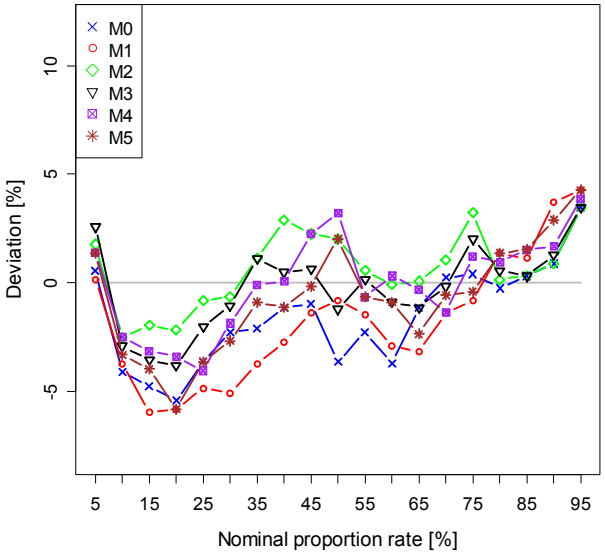

Fig. C-2 Calibration diagram for WFB with 6:00 PM NWP and NW models M0–M5 for look-ahead time step t+15h.

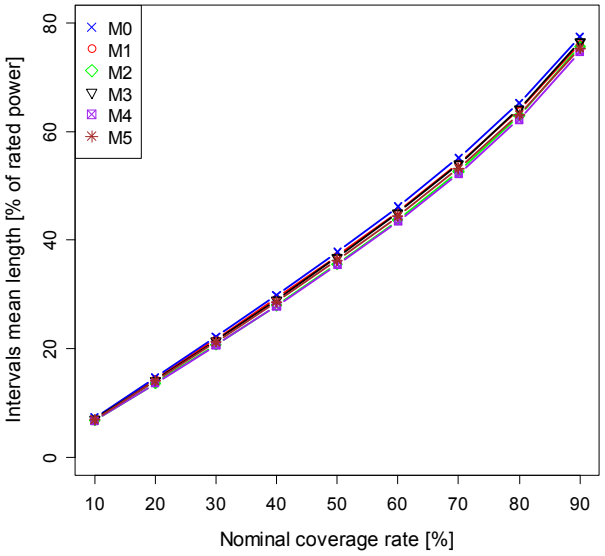

Fig. C-3 Sharpness diagram for WFB with 6:00 AM NWP and NW models M0–M5 for look-ahead time step t+15h.

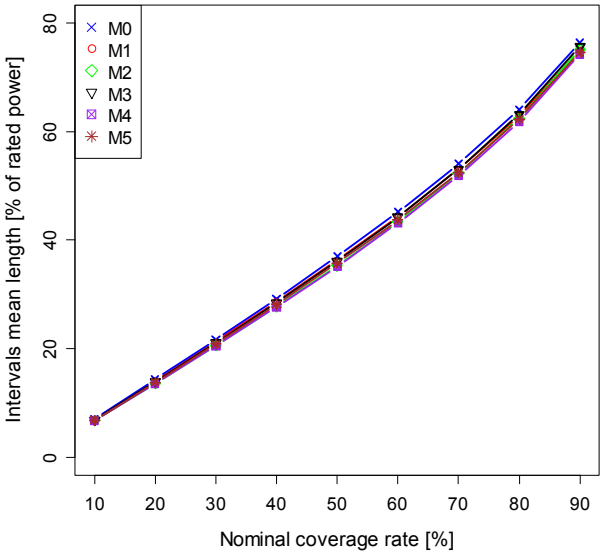

Fig. C-4 Sharpness diagram for WFB with 6:00 PM NWP and NW models M0–M5 for look-ahead time step t+15h.

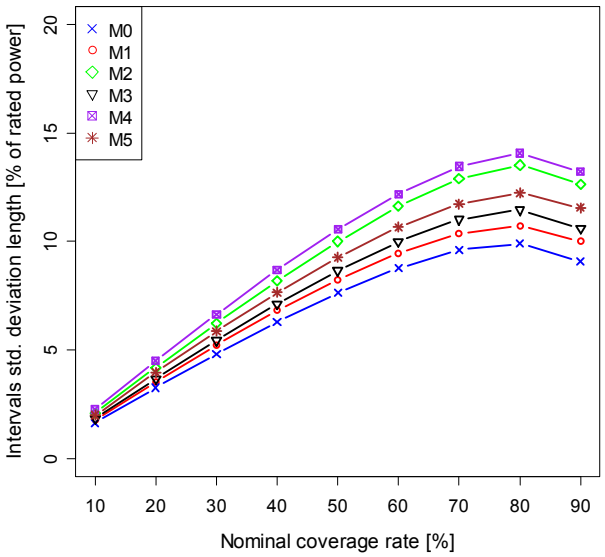

Fig. C-5 Resolution diagram for WFB with 6:00 AM NWP and NW models M0–M5 for look-ahead time step t+15h.

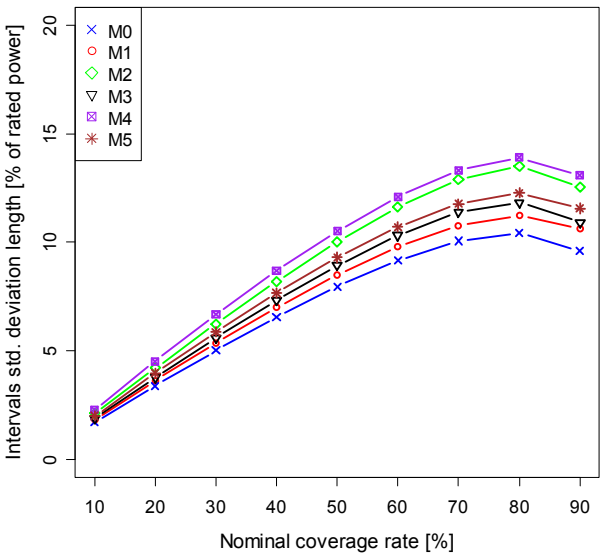

Fig. C-6 Resolution diagram for WFB with 6:00 PM NWP and NW models M0–M5 for look-ahead time step t+15h.

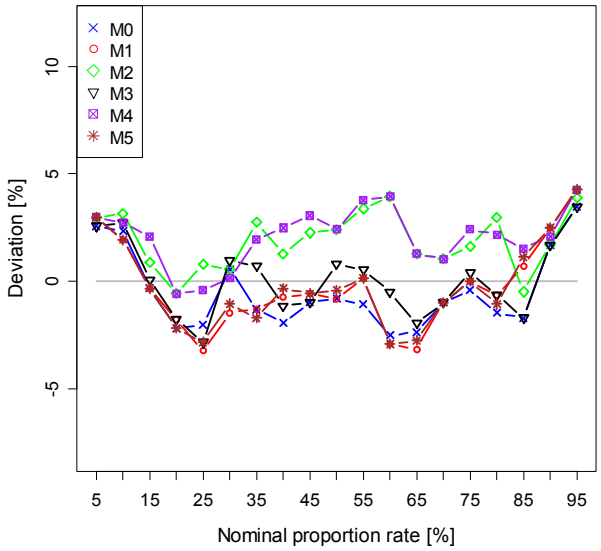

Fig. C-7 Calibration diagram for WFB with 6:00 AM NWP and QC models M0–M5 for look-ahead time step t+15h.

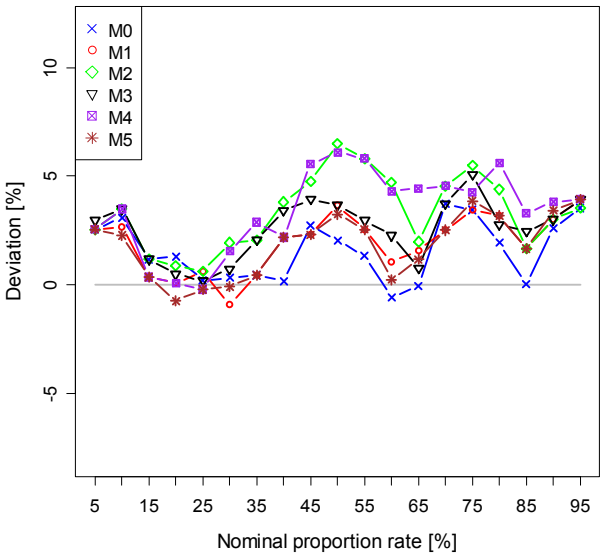

Fig. C-8 Calibration diagram for WFB with 6:00 PM NWP and QC models M0–M5 for look-ahead time step t+15h.

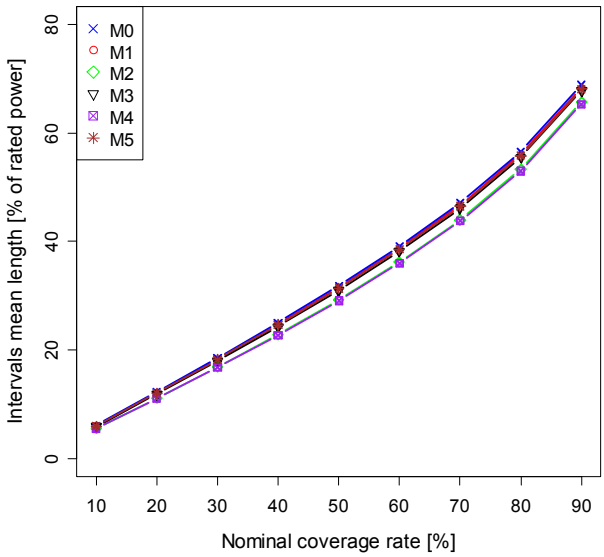

Fig. C-9 Sharpness diagram for WFB with 6:00 AM NWP and QC models M0–M5 for look-ahead time step t+15h.

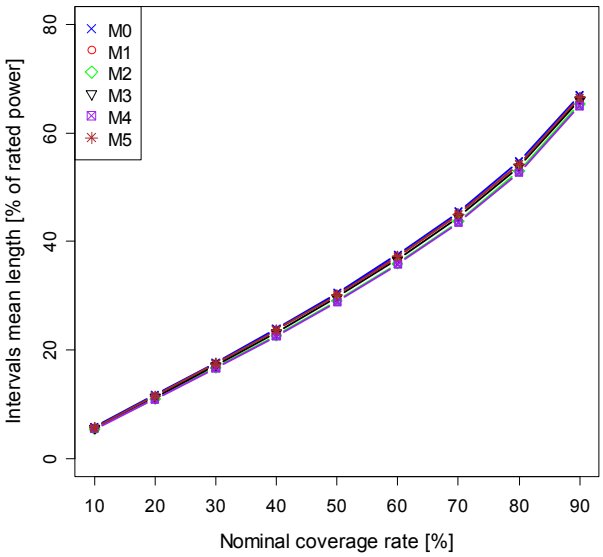

Fig. C-10 Sharpness diagram for WFB with 6:00 PM NWP and QC models M0–M5 for look-ahead time step t+15h.

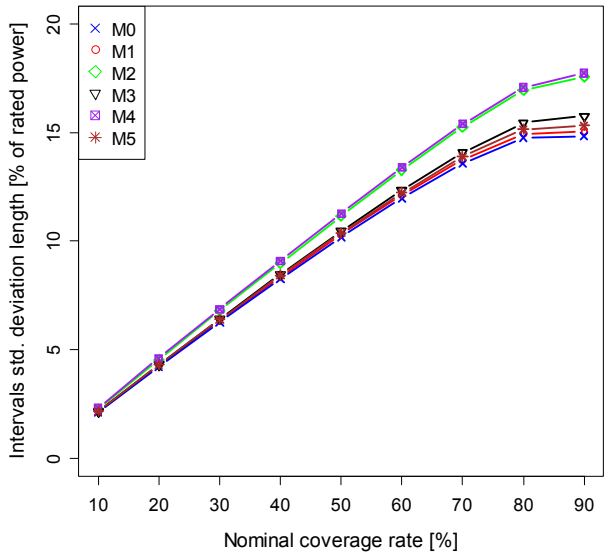

Fig. C-11 Resolution diagram for WFB with 6:00 AM NWP and QC models M0–M5 for look-ahead time step t+15h.

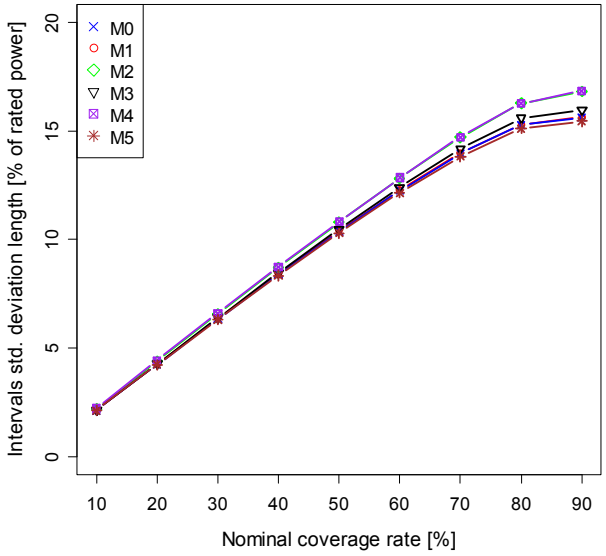

Fig. C-12 Resolution diagram for WFB with 6:00 PM NWP and QC models M0–M5 for look-ahead time step t+15h.

This page intentionally blank

APPENDIX D – TIME-ADAPTIVE EVALUATION RESULTS FOR WIND FARM A

Figs. D-1 through D-4 present time-adaptive results for Wind Farm A.

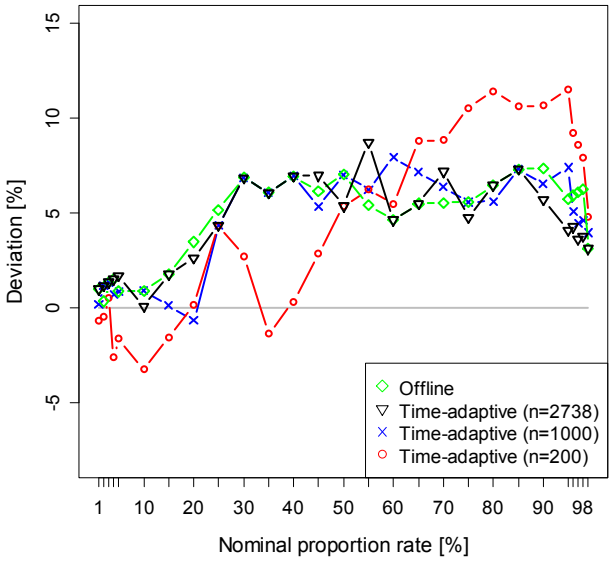

Fig. D-1 Calibration diagram for t+15h obtained with the NW time-adaptive model for the WFA dataset.

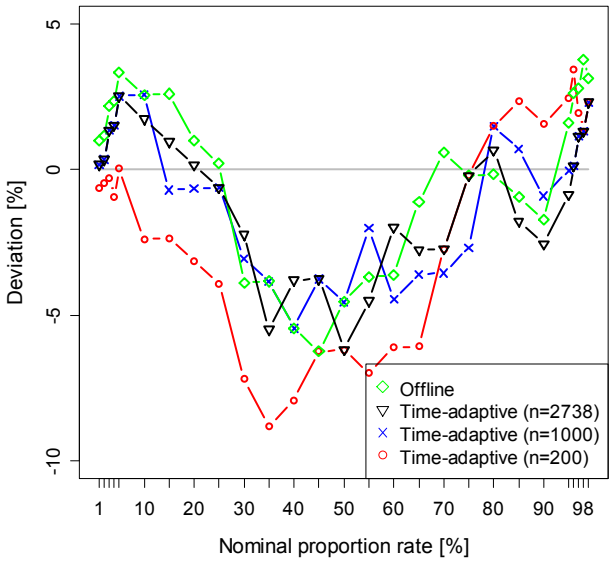

Fig. D-2 Calibration diagram for t+10h obtained with the NW time-adaptive model for the WFA dataset.

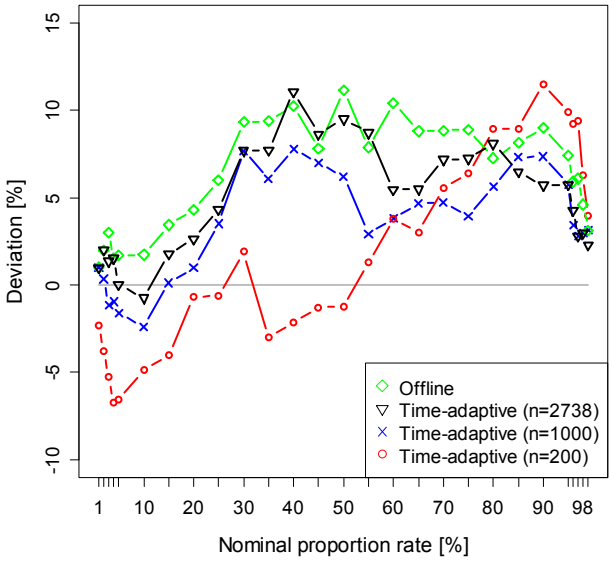

Fig. D-3 Calibration diagram for t+15h obtained with the QC time-adaptive model for the WFA dataset.

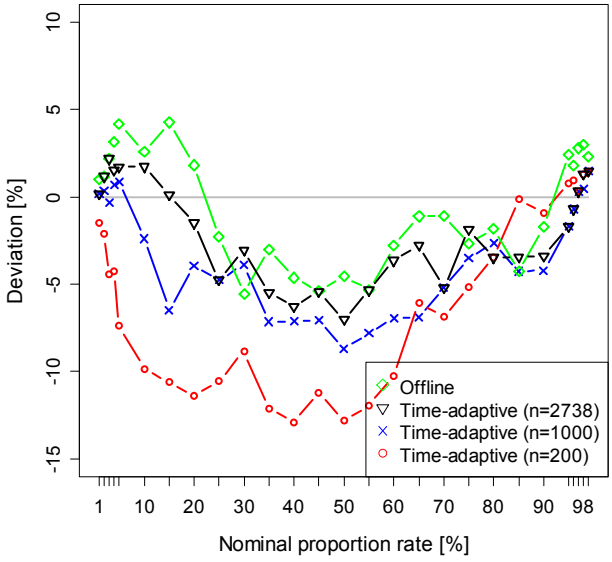

Fig. D-4 Calibration diagram for t+10h obtained with the QC time-adaptive model for the WFA dataset.

This page intentionally blank

APPENDIX E – TIME-ADAPTIVE EVALUATION RESULTS FOR WIND FARM B

Figs. E-1 through E-4 present time-adaptive results for Wind Farm B.

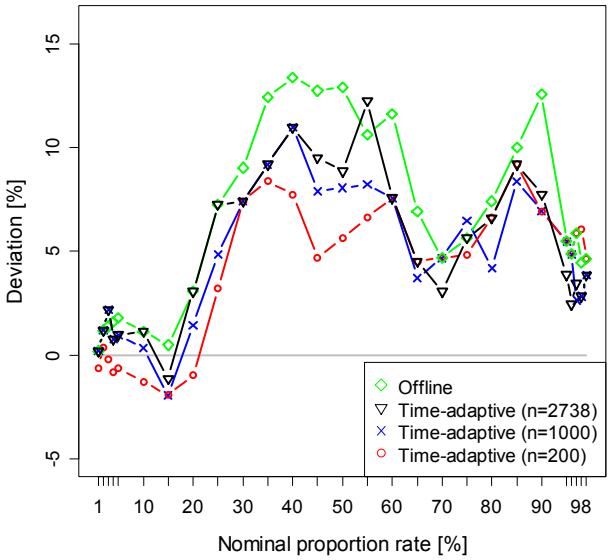

Fig. E-1 Calibration diagram for t+15h obtained with the NW time-adaptive model for the WFB dataset.

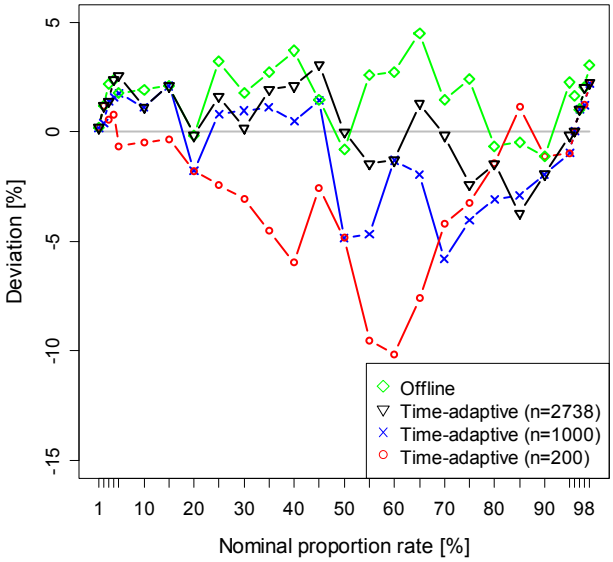

Fig. E-2 Calibration diagram for t+10h obtained with the NW time-adaptive model for the WFB dataset.

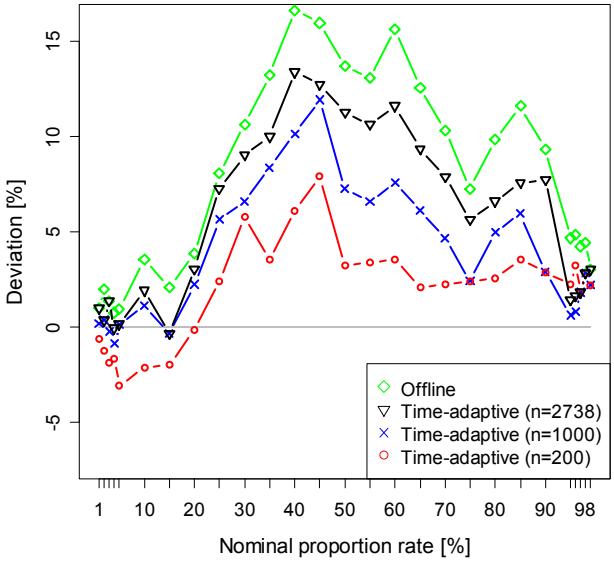

Fig. E-3 Calibration diagram for t+15h obtained with the QC time-adaptive model for the WFB dataset.

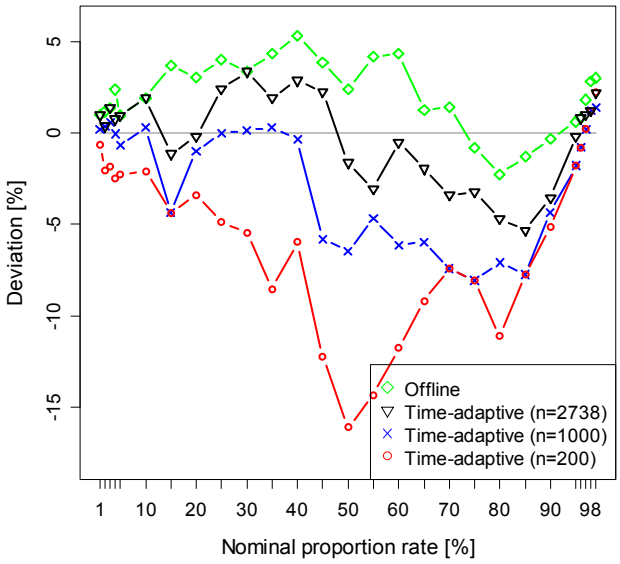

Fig. E-4 Calibration diagram for t+10h obtained with the QC time-adaptive model for the WFB dataset.

This page intentionally blank

Decision and Information Sciences Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 221
Argonne, IL 60439-4844

www.anl.gov