Technical Memo

ECMWF
European Centre for Medium-Range Weather Forecasts

896

Statistical modelling of 2m temperature and 10m wind speed forecast errors

Zied Ben Bouallègue, Fenwick Cooper, Matthew Chantry, Peter Düben, Peter Bechtold, Irina Sandu (Forecast & Research Department)

April 2022 - Manuscript submitted to MWR

Series: ECMWF Technical Memoranda

A full list of ECMWF Publications can be found on our web site under:
http://www.ecmwf.int/en/publications/

Contact: library@ecmwf.int

European Centre for Medium Range Weather Forecasts, Shinfield Park, Reading, RG2 9AX, UK

Abstract

Based on the principle "*learn from past errors to correct current forecasts*", statistical postprocessing consists in optimizing forecasts generated by numerical weather prediction (NWP) models. In this context, machine learning (ML) offers state-of-the-art tools for training statistical models and making predictions based on large datasets. In our study, ML-based solutions are developed to reduce forecast errors of 2m temperature and 10m wind speed of the ECMWF's operational medium-range high-resolution forecasts produced with the Integrated Forecasting System (IFS). IFS forecasts and other spatio-temporal indicators are used as predictors after careful selection with the help of ML interpretability tools. Different ML approaches are tested: linear regression, random forest decision trees, and neural network. Statistical models of systematic and random errors are derived sequentially where the random error is defined as the residual error after bias correction. In terms of output, bias correction and forecast uncertainty prediction are made available at any point-locations around the world. All 3 ML methods show similar ability to capture situation-dependent biases leading to noteworthy performance improvements (between 10% and 15% improvement in terms of root-mean-square error for all lead times and variables), and similar ability to provide reliable uncertainty predictions.

1 Introduction

Near-surface temperature and wind speed are key variables in many weather applications, but numerical weather prediction (NWP) systems struggle in producing bias-free forecasts of such quantities, even at short lead times. In particular, long-standing biases affect the operational medium-range forecasts of 2m temperature and 10m wind speed produced with the Integrated Forecasting System (IFS) of the European Centre for Medium-Range Forecasts (ECMWF), as illustrated in Figure 1.

Recent investigations of ECMWF's near-surface forecast biases shed new light on potential sources of forecast errors and paved the way for ongoing and future model developments for the IFS (Sandu *et al.*, 2020). At the same time, statistical postprocessing offers a pragmatic way to correct systematic errors. By comparing forecasts with in-situ observations, statistical models learn from past errors to derive corrections to be applied to future forecasts. Hemri *et al.* (2014) showed that the expected benefit of postprocessing does not vary year after year, suggesting that benefits from postprocessing and benefits from NWP model improvements are complementary.

In this study, statistical postprocessing of IFS forecasts is investigated with a focus on the ECMWF"s operational deterministic high-resolution forecasts of 2m temperature and 10m wind speed. More specifically, we assess and predict systematic and residual forecast errors using machine learning (ML) tools. The following semantics is used throughout the text: *systematic errors* refer to differences between forecasts and observations that can be corrected for by postprocessing through *bias correction*, while *residual errors* refer to the remaining forecast errors after bias correction.

ML provides a general framework for applying complex statistical methods to large datasets that finds natural applications in the postprocessing of weather forecasts (Düben *et al.*, 2021). Recent developments of ML software libraries such as scikit-learn in Python programming language (*https://scikit-learn.org*) greatly facilitate the take-on of state-of-the-art ML methods. Moreover, advances in ML interpretability (McGovern *et al.*, 2019) provide suitable tools to initiate positive feedback loops between NWP model developers and postprocessing experts. Here, our ML-based postprocessing applications intend to:

- capture bias patterns and estimate forecast uncertainty,

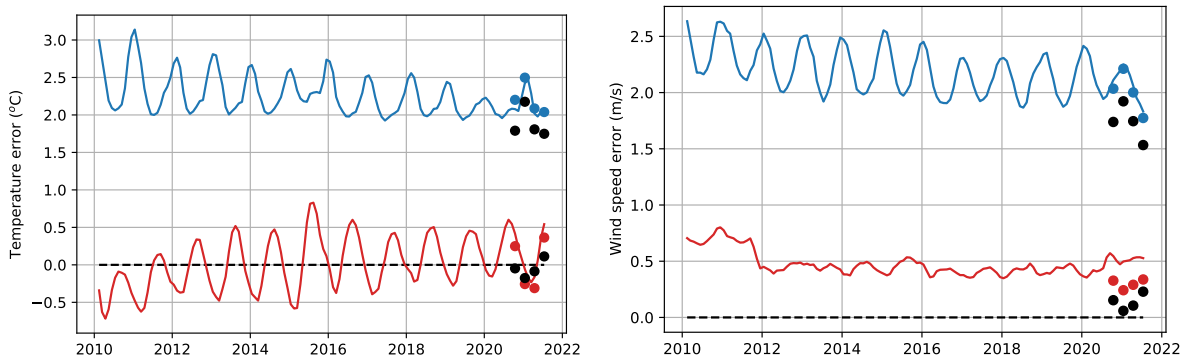

Figure 1: 48 hour forecast performance over Europe ($35^oN - 75^oN$, $12.5^oW - 42.5^oE$) of IFS 2m temperature (left) and 10m wind speed (right) over the last decade, for ECMWF's operational high-resolution forecasts with respect to synop observations. The blue (root-mean-square error) and red (mean error) lines indicate the calculation as performed by Haiden et al. (2021), while the respective dots represent the equivalent calculation performed with the data quality control criteria used here (see text). The black dots indicate the resulting errors after postprocessing by the best ML-based models derived here (see text for details).

- compare the performance of different postprocessing methods,
- help identify sources of errors,

in the context of global forecasting of surface weather variables.

Postprocessing of global forecasts requires large datasets in order to provide relevant contextual information about the forecast to be corrected. So-called *predictors* help distinguish between different situations (in a static or a dynamic sense) leading, on average, to over- or under-prediction, and, on average, to a large or a small forecast error. The general strategy consists in including a variety of predictors as input to the ML models: NWP model output (such as the forecast surface pressure), model characteristics (such as the model orography), and spatio-temporal indicators (such as the day of the year). ML algorithms are designed to find useful relationships between *the predictand* (here the forecast error) and the diverse sets of predictors. The use of such ML approaches for successful weather forecasting applications have been documented in recent years:

1. linear regression techniques for the postprocessing of ensemble solar radiation forecasts over Germany (Ben Bouallègue, 2017),
2. decision trees from random forests for the postprocessing of temperature and wind speed forecast over France (Taillardat *et al.*, 2016),
3. neural networks for the postprocessing of ensemble temperature forecasts over Germany (Rasp and Lerch, 2018),

to cite a few examples in an effervescent field of research. The interested reader can find an overview of postprocessing techniques and recent developments in this research area in Vannitsem *et al.* (2021).

Here, we propose to test and compare 3 statistical methods: linear regression (LR), random forests (RF), and neural networks (NN). The goal is to provide statistically postprocessed forecasts at any location over the globe based on 2m temperature and 10m wind speed IFS forecasts. In contrast with previous

studies, systematic and residual errors are treated sequentially rather than at the same time. Additionally, we explore the benefit and impact of using postprocessing configurations where input data consists of static predictors (*e.g. non-state-dependent) and time indicators only.* Finally, following a suggestion in Hamill (2021), the combination of the different statistical models is also tested.

The remaining of the manuscript is organised as follows: Section 2 details the data used in this study, the statistical models are described in Section 3, the selection of predictors in Section 4. The results are presented and discussed in Section 5 before concluding in Section 6.

2 Data

2.1 Forecasts, observations and predictors

The forecasts of 2m temperature and 10m wind speed used in this study are the operational ECMWF high-resolution ($\sim$ 9 km) ten-day global weather forecasts produced with ECMWF IFS (ECMWF, 2020). The data is taken over two years (Sept. 2019 to Aug. 2021) from forecasts starting each day at 00:00 and with lead times up to 48 hours, at 3-hour intervals.

Observations are measurements at synoptic weather stations (SYNOP) received through the World Meteorological Organization (WMO) Global Telecommunications System* (GTS). For each weather station, the nearest neighbour 2m temperature and 10m wind are taken from the forecast at the nearest neighbouring point of a measurement station.

There is a difference between the height of model orography at a station location and the true height of the station. When comparing the forecasts and observations, the standard 2m temperature forecast correction corresponds to a linear reduction in temperature with height (a lapse rate) of 6.5^{o}C km^{-1} while taking the nearest neighbouring point in the model grid as the model elevation. This approach is considered as the ***default*** bias correction in the following.

We consider a variety of potential predictors for our ML experimentations. The full list of predictors is provided in Table 2.1. We test 2 types of model configurations:

1. a "*state-dependent*" configuration where the current forecast and any other model output can be used as a predictor (*i.e.* there is no self-imposed restrictions on the use of predictors),

2. a "*state-independent*" configuration where only predictors available before the start of the forecast-of-the-day are used.

We distinguish 3 types of predictors: state-dependent predictors which are direct model outputs that differ for each forecast, static predictors which describe constant characteristics of the model surface, and time indicators (see the classification in Table 2.1). In a *state-independent* configuration, input data only include static predictors (such as the model orography) and time indicators (such as the day of the year) offering a 3D† model of the forecast errors independent of the forecast-of-the-day.

*Unfortunately, we cannot make our dataset publicly available because of restrictions regarding the redistribution of SYNOP data.

†2D in space plus the time dimension

	2m temperature		10m wind speed	
Predictors / Configurations	state-dependent	state-independent	state-dependent	state-independent
State-dependent predictors				
2m temperature (IFS)	○	·	·	·
10m wind speed (IFS)	·	·	○	·
2m dewpoint	○	·	–	·
Skin temperature	○	·	–	·
Lowest model level temperature	○	·	○	·
Lowest model level wind speed	○	·	–	·
Lowest model level meridional wind	–	·	○	·
Lowest model level zonal wind	–	·	○	·
Second level meridional wind	·	·	–	·
Second level zonal wind	·	·	–	·
Total cloud cover	–	·	–	·
Low cloud cover	–	·	–	·
Snow depth	○	·	–	·
Surface solar radiation flux	○	·	○	·
Surface thermal radiation flux	○	·	–	·
Latent heat flux	–	·	–	·
Sensible heat flux	○	·	–	·
Top soil layer temperature	○	·	–	·
Top soil layer frozen (1) or not (0)	○	·	–	·
Boundary layer height	○	·	○	·
CAPE	–	·	–	·
Volumetric soil water layer	–	·	–	·
Surface pressure	○	·	–	·
Aerodynamic roughness length	–	·	–	·
Static predictors				
Model orography	–	–	–	–
Slope of sub-grid orography	–	–	–	–
Standard deviation of sub-grid orography	○	○	○	○
Land sea mask	○	○	○	○
Soil type	–	–	–	–
Vegetation cover low	–	○	○	○
Vegetation cover high	–	○	○	○
Vegetation type low	–	○	–	–
Vegetation type high	–	○	–	–
Latitude	–	○	○	○
Longitude	–	–	○	○
Cos(longitude)	–	○	○	○
Station elevation	○	○	–	–
Station-Model elevation	○	○	○	○
Log(station-model) elevation	○	○	○	○
Time indicators				
Day of year	–	–	–	–
Cos(day of year)	–	○	–	○
Sin(day of year)	○	○	–	–
Local time of day	○	○	–	–
Cos(time of day)	○	○	–	○
Solar zenith angle	–	○	–	–
Start date	–	–	–	–
Forecast lead time	–	–	–	–

Table 1: Predictors list, classification, and use in different configurations: [○] selected ,[−] not selected, [·] not tested.

2.2 Observation quality control

Observation quality control is first based on observation meta-data. To start with, the elevation of each station is not necessarily fixed over the two-year period. Sometimes estimates of its location change, perhaps by rounding errors in the reported latitude and longitude. This changes the station's elevation if it is automatically read from a map. Sometimes a station's altitude is measured differently and sometimes the station actually moves. If a station moves, its elevation doesn't necessarily change, but the model elevation might. 1882 of 13573 (14%) stations exhibit elevation changes.

Sometimes, for some of the measurements at a particular station, the elevation is not recorded. In this case, we set the station elevation for the purposes of modelling to an elevation that is recorded for that station, before applying the criteria below.

We adopt the following WMO control criteria (WMO, 2019). Independently for wind and temperature observations, measurements are rejected if:

- The surface pressure is higher than 700 hPa (low elevation) and the measured vs. lapse rate corrected forecast 2m temperature difference is more than 15^{o}C.
- The surface pressure is lower than 700 hPa (high elevation) and the measured vs. lapse rate corrected forecast 2m temperature difference is more than 10^{o}C.
- Over the ocean the mean difference between the measured 2m temperature and the lapse rate corrected forecast temperature is greater than 4^{o}C.
- Over the ocean the standard deviation of the difference between the measured 2m temperature and the lapse rate corrected forecast temperature is greater than 6^{o}C.
- Over the ocean the mean difference between the measured 10m wind and the forecast wind is greater than 5ms^{-1}.
- The surface pressure is higher than 775 hPa (low elevation) and the measured vs. forecast 10m wind difference is more than 35ms^{-1}.
- The surface pressure is between 775 hPa and 600 hPa (middle elevation) and the measured vs. forecast 10m wind difference is more than 40ms^{-1}.
- The surface pressure is lower than 600 hPa (high elevation) and the measured vs. forecast 10m wind difference is more than 45ms^{-1}.

Contrary to the WMO quality control criteria, 10m wind measurements were not rejected if there were less than 10 measurements in any particular month, or if the RMS forecast error that month is greater than 15ms^{-1}. In addition, measurements are rejected for:

- All stations (126) that moved more than 10km.
- All forecasts where the observation latitude or longitude changed at all during the validity time of the forecast.
- All stations with elevations recorded above 10,000m, or stations where the elevation is never recorded.

- Four predictors are derived from heat and radiation fields: surface solar radiation, surface thermal radiation, sensible heat, and latent heat fluxes. These are stored as cumulative quantities. They are converted to instantaneous fluxes using first-order finite differences. Where there is a gap between 2m temperature or 10m wind measurements of 6 hours or more, the finite difference approximation is not sufficiently accurate and the forecast at this station location is rejected.
- All forecasts where we don't have an initial measurement at zero lead time.
- All stations recording a 2m temperature above 56.7°C (207 stations) or below -78°C (73 stations). (Only 2m temperature measurements rejected.)
- All stations (1684 stations) recording a 10m wind speed above 50m/s (180 km/h). (Only 10m wind measurements rejected.)

2.3 Training, verification and test data

We use data over the two-year period 1 September 2019 - 31 August 2021 and the data is split into three segments: training, verification, and test. The training data is the portion of the data that the ML-based models are fit to. We use 1 year for the training data, 1 September 2019 - 31 August 2020 and half of the stations (even-numbered, with numbers randomly attributed). The verification data is the portion of the data that is reserved for optimisation of all free model parameters, often called hyper-parameters. For example, we don't know the number of trees to use in a random forest, the number of neurons to use in a neural network or the step size in an iterative descent. We also use even-numbered stations for the verification data. The test data is reserved for the end to finally test model predictions of data that they have not yet seen. We conduct 4 experiments with different validation and test data sets:

Experiment	Verification	Test
1	March-April-May 2021	September-October-November 2020
2	June-July-August 2021	December-January-February 2020-2021
3	September-October-November 2020	March-April-May 2021
4	December-January-February 2020-2021	June-July-August 2021

In addition to the split as a function of the date, test data is taken from odd-numbered stations while even-numbered stations are used for training and verification. This scheme ensures that there is no overlap between training/verification and test data. When discussing the results in Section 5, we focus on a summer season (experiment 2) and a winter season (experiment 4) only.

3 ML-based Models

We want to model the difference between a forecast denoted f, and the corresponding measurement at a weather station denoted o. We consider 3 ML models and, for each model, 2 configurations based on the chosen pool of predictors, and finally a combination of models for each configuration. More explicitly, we perform the following for 2m temperature and 10m wind independently:

1. Test 3 types of ML methods: linear regression (***LR***), random forests (***RF***) and neural networks (***NN***), see sections 3.1, 3.2 and 3.3, respectively.

2. For each method, consider 2 model configurations depending on the pool of predictors we select from: a ***state-dependent*** configuration and a ***state-independent*** configuration, see Section 2.1.
3. For each configuration, select a ***subset of predictors*** (optimised by trial and error testing and with the help of ML interpretability tools that assess predictors importance) and use this subset consistently with all 3 ML methods, see section 4.1.
4. For each method and configuration, fit 2 models to predict the ***systematic error*** and the ***residual error*** separately: "Model 1" fitted to the raw forecast error $f-o$ and "Model 2" fitted to the remaining error $(\hat{f}-o)^2$ after bias correction with $\hat{f}$ the bias-corrected forecast, see Figure 2.
5. For each method and configuration, take the average of the 3 ML predictions to make a ***combined*** prediction.

In our approach, we use 2 distinct statistical models for the representation of systematic errors on the one hand and the representation of residual errors on the other hand. Model 1 focuses on the raw forecast error denoted e and defined as $e = f - o$. The estimated systematic error, *i.e.* the output of Model 1, is denoted $\hat{e}$. Model 2 focuses on the residual error denoted ξ and defined as $\xi = (\hat{e} - e)^2$. The residual error can be expressed as the squared difference between corrected forecast and observation:

$$\xi = (\hat{e} - e)^2 \tag{1}$$
$$= (\hat{e} - (f - o))^2 \tag{2}$$
$$= (\hat{f} - o)^2 \tag{3}$$

where $\hat{f} = f - \hat{e}$ is the bias-corrected forecast. Using synthetic data, a simple example of how Model 1 and Model 2 works in practice is provided in Fig. 2.

The choice of a 2-model approach is motivated by the fact that no assumptions about the form of the underlying forecast probability distribution are required in that case. Such assumptions are for example required when using parametric methods which target the optimisation of a probabilistic score such as the continuous ranked probability score (CRPS). Moreover, with our 2-model approach, ML interpretability tools can be beneficially applied to each model (Model 1 and Model 2) separately as illustrated below in Section 4.2. With traditional non-parametric methods relying on analog forecasts, for example, it is difficult to distinguish between sources of different error types (systematic and residual).

Besides the standard *state-dependent* configuration, we also test here a *state-independent* configuration for each model. The idea is sparked by the ML interpretability results indicating that static predictors and time indicators are "important" features, *i.e.* among top-ranked predictors, in particular for 10m wind speed predictions (see again Section 4.2). In addition, the foreseen advantages of using *state-independent* configurations in research or operational settings are multiple: building large training datasets is simple and fast, no critical time processing is involved as all operations are performed offline, the possibility to check the statistical model output before dissemination is an asset. Besides, when using a *state-independent* configuration, the estimated systematic and residual errors are location and time-specific but independent of the forecast-of-the-day. As an application, the derived model of the forecast-observation discrepancy could for example help detect spurious observations in an enhanced quality control scheme.

The configuration setup can be summarized as follows. The bias-corrected forecast $\hat{f}$ is derived as the difference between the raw IFS forecast f and the error estimate $\hat{e}$ applying one of the 2 following configurations:

$$\text{State-dependent configuration:} \quad \hat{f} = f - \hat{e}(d,s,t) \tag{4}$$
$$\text{State-independent configuration:} \quad \hat{f} = f - \hat{e}(s,t) \tag{5}$$

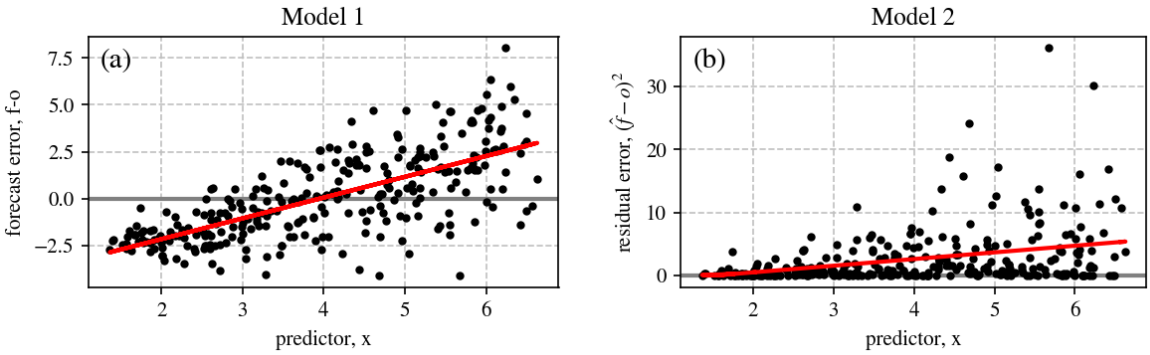

Figure 2: Illustrative example of the problem at hand based on synthetic data. a) We are first interested in predicting the forecast error $e = f - o$ as a function of a predictor x. Model 1 provides the best estimate of the error as a function of x as represented by the solid red line. The residual error corresponds to the squared distance between the red line and the black dots. b) As a second step, Model 2 is built to capture the residual error $e = (\hat{f} - o)^2$ as a function of x. The resulting estimated residual error is represented by the red line. In this simple example, Models 1 and 2 are both linear regression fits with 2 parameters each.

where d are state-dependent predictors, s static predictors, and t time indicators. In both cases, the bias-corrected forecast is a function of the IFS forecast of the day, but the error correction part is weather dependent only in the *state-dependent* configuration. The exact form of the function $\hat{e}(\cdot)$ depends on the ML method applied as described below.

3.1 Linear regression

To fit a function using linear regression, the functional form of the fit coefficients that we are trying to find must be linear. For example, consider

$$\hat{e}(x) = c_0 + c_1 x + c_2 x^2.$$

We have a list of values of x, the "predictor", and a list of values of e, the "predictand" and we want to find the values of the fit coefficients, c_0, c_1 and c_2, also known as the "parameters". The function itself is non-linear due to the x^2 term, but each term is linear in the fit coefficients. Given values of x and $e(x)$, linear regression then finds the fit coefficients that minimise the squares of the differences between the function $\hat{e}(x)$ and the values of e provided. See for example Press *et al.* (2007) for a more detailed explanation.

Here we consider quadratic functions of our predictors. With two predictors x and y and quadratic terms, a quadratic model takes the form:

$$\hat{e}(x,y) = c_0 + c_1 x + c_2 y + c_3 x^2 + c_4 xy + c_5 y^2$$

with 6 unknown parameters to fit in this simple example. The precise number of unknown parameters depends on the number of predictors, for example 66 parameters with 10 predictors, 231 parameters with 20 predictors, and so on.

3.2 Random forest

Random forest (RF) is a non-parametric technique that consists in building a collection of trees. Each tree is a decision tree that partitions a multidimensional dataset into successively smaller subdomains. Each partition of the data consists in splitting the data into two groups based on some threshold applied to one of the predictors. Predictors and thresholds are chosen in order to maximize the diversity of the response variable among the resulting groups. Each new group is itself split into two, and so on until some stopping criterion is reached. In a prediction situation, the current values of the predictors draw a path in the tree until a final leaf. The forecast takes the mean value of the response variable in the final group (leaf).

Our implementation makes use of the *scikit-learn* Python library (Pedregosa *et al.*, 2011). In the interest of computational time, training is performed on a subsample of the dataset by randomly selecting 1% of the total available training data. The models for the prediction of systematic and residual errors are trained on 2 different randomly selected subsamples. The main hyper-parameters associated with the RF models are the number of trees and the maximum depth of each tree. The hyper-parameters selected for the different experiments presented in this study are shown in Table 2.

	2m temperature		**10m wind speed**	
	state-dependent	state-independent	state-dependent	state-independent
number of trees	200	100	100	50
maximum depth of each tree	25	13	15	10

Table 2: Hyper-parameters settings for the different random forest models.

3.3 Neural network

Here we use a multi-layer perceptron (MLP), also known as a fully-connected neural network as our neural network design. We choose 4 hidden layers, with 32 hidden neurons, resulting in approximately 4000 trainable parameters (the precise number depends on the number of predictors used). For hidden layers, we use the Swish activation function, for output layers we use no activation function. We build and train these models using Tensorflow/Keras. Models are fitted using the Adam optimizer, with a learning rate of 10^{-3}. We train to minimise the mean-squared error for 20 epochs (passes through the training set) with a batch size of 128. Early stopping, after the validation loss has failed to decrease for 6 epochs, and learning rate reduction (again based on the validation loss) are also employed, but the results were not found to be sensitive to these choices. We also explored increasing the number of trainable parameters, through increases in hidden neurons and hidden layers, but these increases did not return a noticeable reduction in losses on the testing dataset.

4 Predictor selection and ranking

4.1 Backwards stepwise elimination

To select predictors we use backwards stepwise elimination and linear regression of quadratic polynomials. The algorithm proceeds as follows:

1. Each predictor is removed from the full list of n predictors one at a time and the regression is performed. We then have n regression models each fitting $n-1$ predictors.

2. Each of these regression models is then used to forecast the training data. The predictor that corresponds to the smallest reduction in RMSE between the model and the training data, is discarded. The list of predictors is then one shorter than when we started.

3. The entire procedure is then repeated to find and remove the next least "important" predictor. This is repeated until there is one predictor left.

Applying the resulting models to predict the validation data indicates that backwards elimination is sufficient, see Fig. 3. Note that if two predictors both contribute the same information, removing either one will not reduce the RMSE, and one of the two will be randomly selected as unimportant. The algorithm only considers it "important" to include one of these two predictors in a model. Some of our predictors are highly correlated, for example, the skin temperature and the temperature on the lowest model level, at the station locations, have a correlation coefficient of 0.97 over the training data set.

Results in Fig. 3 serve as a basis for predictor selection of the models with no restrictions in the choice of predictors (contrary to the *state-independent* configurations). As a complementary tool, RF impurity importance, as discussed below, is also explored. Eventually, the final set of predictors is selected scrutinizing ML interpretability plots with a critical (human) eye. For example, wind speed components at the lowest model level are ranked poorly for 2m temperature predictions in Fig. 3 but we included them in the list because they are considered important by RF models. Also, the land-sea mask is added to our list as deemed important from a practical point of view. The list of selected predictors for each configuration (state-dependent and *state-independent*) and for each variable (2m temperature and 10m wind speed) is detailed in Table 2.1.

4.2 Random forest impurity importance

The interpretability of RF models is facilitated by the so-called *feature importance* results. Indeed, RF algorithm allows identifying the more valuable predictors in the process of building decision trees. Predictor importance is measured by the *mean decrease in impurity* where *impurity* is measured by the Gini coefficient (a metric proportional to the area under the relative operating characteristics curve). The *mean decrease in impurity* corresponds to the total decrease in node impurity averaged over all trees (Louppe *et al.*, 2013). It is worth noting that this measure favours predictors with high cardinality, i.e. predictors with many unique values. In our case, many of our predictors take a lower number of unique values than others (*e.g.* elevation is constant at each station). For this reason, backwards stepwise regression is considered as our tool of choice for predictor selection.

Nevertheless, predictor impurity importance is key for the interpretability of RF models. The ranking of the predictors for the models of 2m temperature and 10m wind speed errors is shown in Figure 4 and 5, respectively. Predictor importance for models of systematic errors (Model 1) and of residual errors (Model 2) are provided separately as 2 different RF models are trained consecutively. Our findings are the following:

- For 2m temperature, boundary layer height forecasts, temperature forecasts at various levels in the atmosphere and the ground, and wind speed forecasts play together a key role in estimating 2m temperature systematic errors.

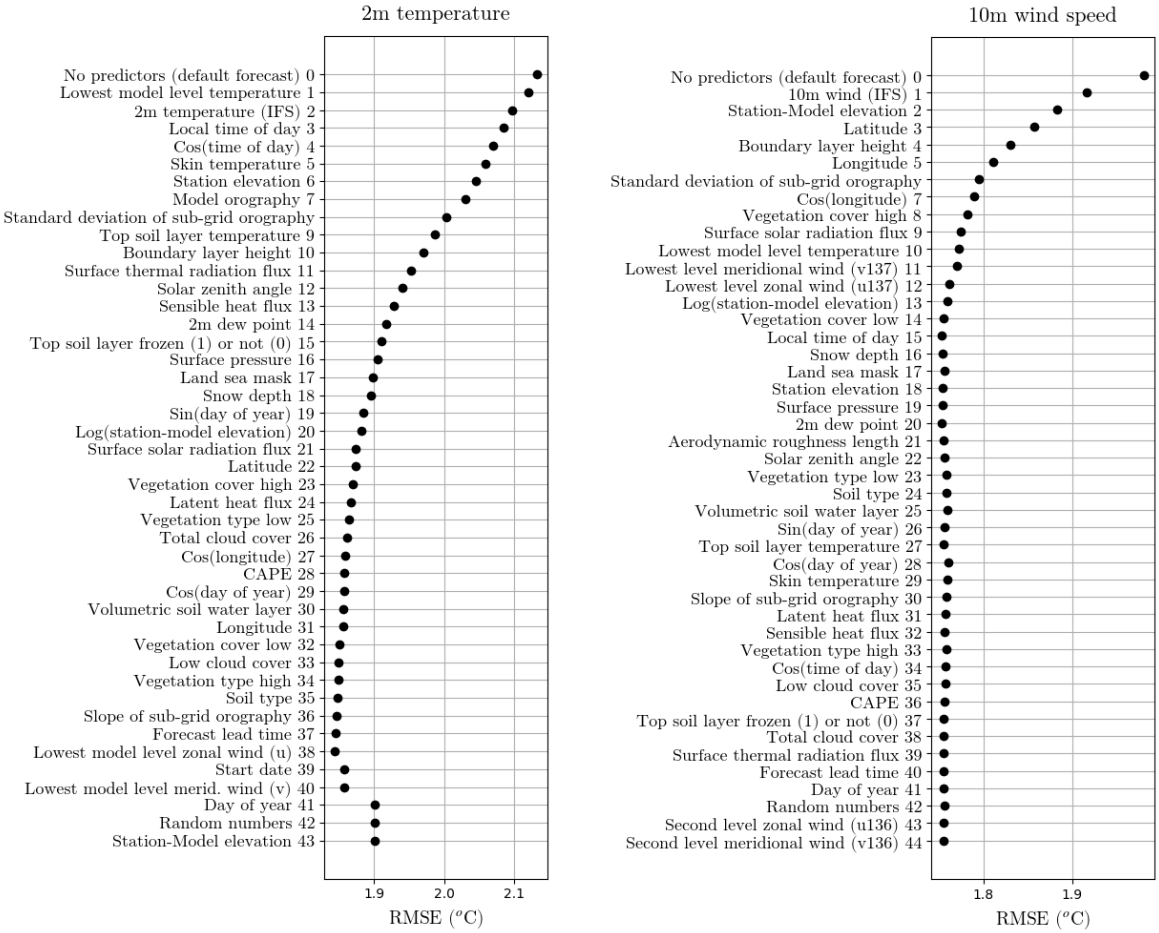

Figure 3: RMSE of 2m temperature (left) and 10m wind speed (right) forecasts (against SYNOP observations) as a function of the number of predictors used. Results for the summer test period only. For each plot, the top point is the RMSE of the default forecast and the following ones after bias correction with a linear regression model with increasing incrementatly the number of predictors. The order of the predictors is obtained using ***backwards stepwise regression*** *(see text).*

- For 10m wind speed, besides predictors related to the wind itself, static predictors and time indicators appear particularly important. Verification results presented in Section 5 confirm that *state-independent* configurations offer competitive solutions for 10m wind speed predictions.

- Predictor importance for residual errors is dominated by one or two predictors, namely the forecast itself and, for 2m temperature, the boundary layer height. These predictors are also important for systematic error prediction. This result suggests that, in a 2-model approach, systematic error prediction could serve as a predictor for the prediction of residual errors (not tested here).

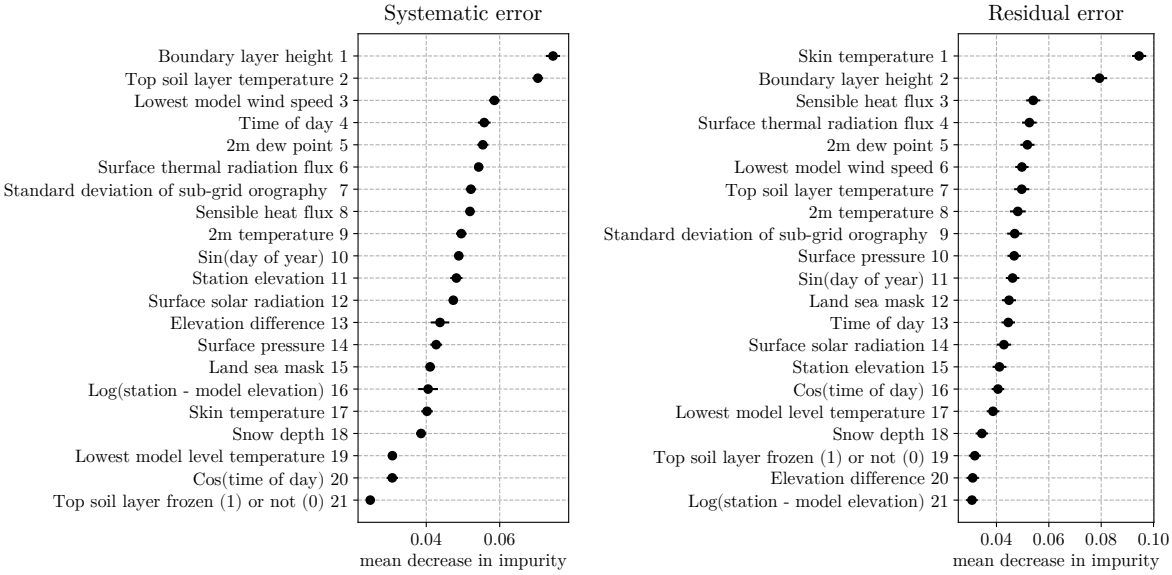

*Figure 4: Predictor **importance** for the prediction of 2m temperature systematic errors (left) and residual errors (right). Importance is estimated with RF mean decrease in impurity. The error bars indicate the inter-trees variability.*

5 Verification results

5.1 Forecast bias

We first focus on the bias to assess the ability of the ML models to correct for systematic errors. Forecast bias (or mean error) is computed as the mean difference between forecasts and observations. We look at two types of score aggregation: one spatial aggregation leading to scoring as a function of the forecast lead time, and one temporal aggregation at each station location. Results of the former are presented in Fig. 6 while results of the latter are presented in Fig. 7 and Fig. 8 for 2m temperature and 10m wind speed, respectively.

In Fig. 6, we compare forecast performance when applying the standard lapse rate correction as described in Section 2.1 (Default), linear regression (LR), random forest (RF), neural network (NN), and a combination of the 3 different ML models (Combined). At this stage we only show results for the *state-dependent* configurations (*i.e.* with no restriction on the choice of predictors). In Figs 7 and 8, the

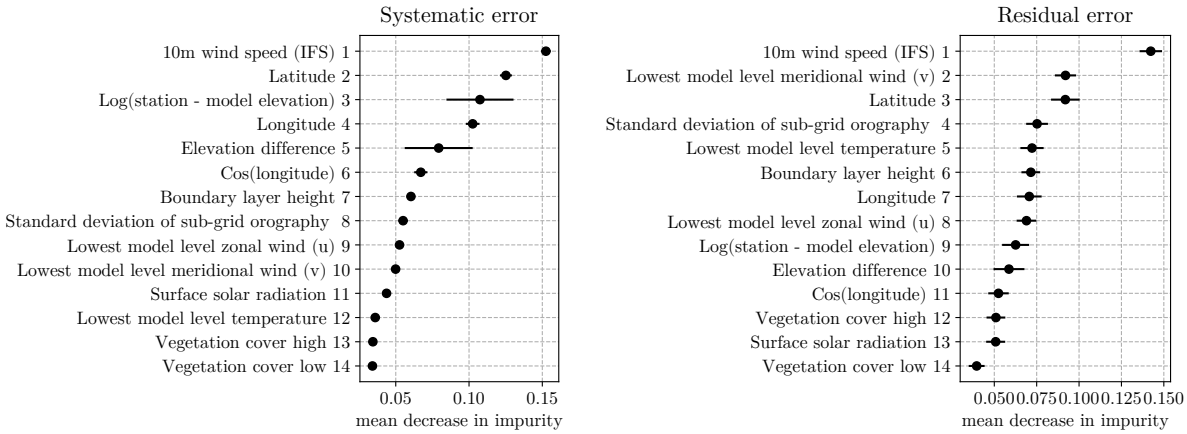

*Figure 5: Predictor **importance** as in Fig. 4 but for 10m wind speed.*

maps focus on the results of the combined models only. For each plot, we distinguish winter and summer results.

The strong diurnal cycle of the bias almost disappears after postprocessing as shown in Fig. 6. The original daily cycle in these plots reflects the daily cycle of the forecast error over Europe where more stations are located. All 3 ML methods perform equally well on average. The bias of 2m temperature forecast seems to slightly increase with lead time in summer. The forecast step could be included as a predictor to capture potential bias drift with forecast horizon if the intention is to apply models trained at short lead times for bias correction of forecasts at longer lead times. Interestingly, the bias of 10m wind speed forecast exhibits the same pattern before and after postprocessing but with a significantly lower amplitude.

At the station level, when looking at each ML model separately, bias correction can perform differently for different models (not shown). The combined model approach benefits from this diversity. Overall, large reductions in forecast bias are visible in various regions of the world for both seasons and weather variables. For 2m temperature in Fig. 7, we note a reduction of the large positive biases dominating the Northern Hemisphere and the negative biases along the Tropics. For 10m wind speed in Fig. 8, we see a clear reduction of the bias in Easter-Europe, over the Indian subcontinent, and the South-American continent.

The distribution of stations is uneven around the world as illustrated in Figs 7 and 8. Data pre-processing, in the form of upscaling, could help homogenise the data before training for systematic errors. In our experiments, ML models are biased towards Europe because of the higher number of station measurements available through GTS in this region. Regions with low data density could benefit from training on a larger area in order to increase the training data sample. However, reversely, training on European stations only improves postprocessed forecasts over Europe but degrades performance on a global scale (not shown).

5.2 Forecast accuracy

Forecast accuracy is mainly assessed with the root-mean-square error (RMSE). In addition to the LR, RF, NN, and combined predictions for the models with no restrictions in terms of predictor choice, we also

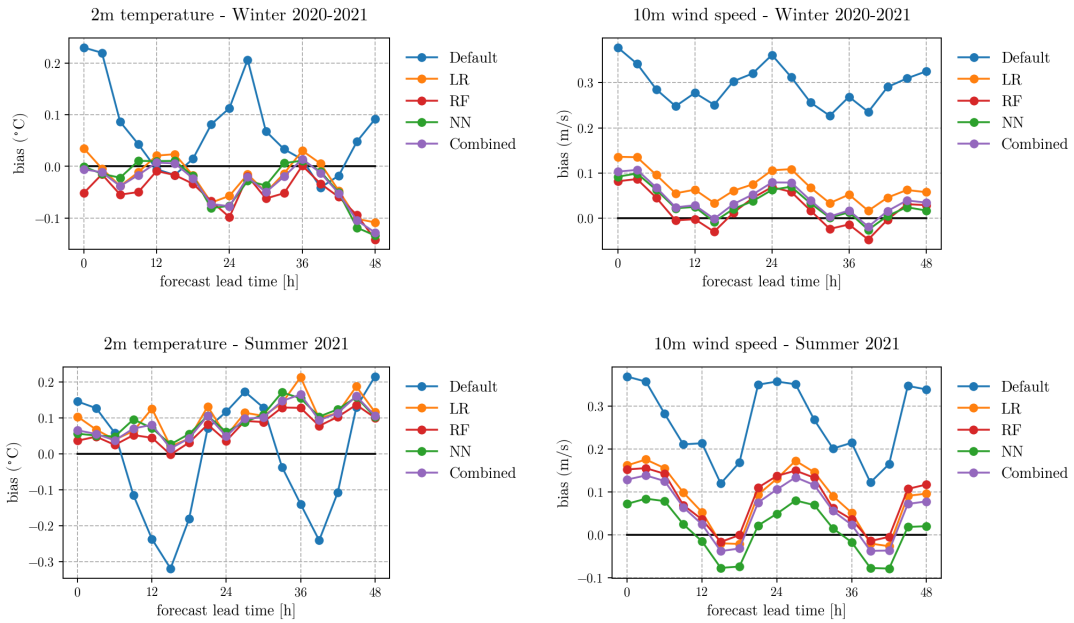

Figure 6: Forecast **bias** *as a function of the forecast lead time. Forecast of 2m temperature (left) and 10m wind speed (right), in winter (top) and in summer (bottom). Zero bias is indicated by a black horizontal line.*

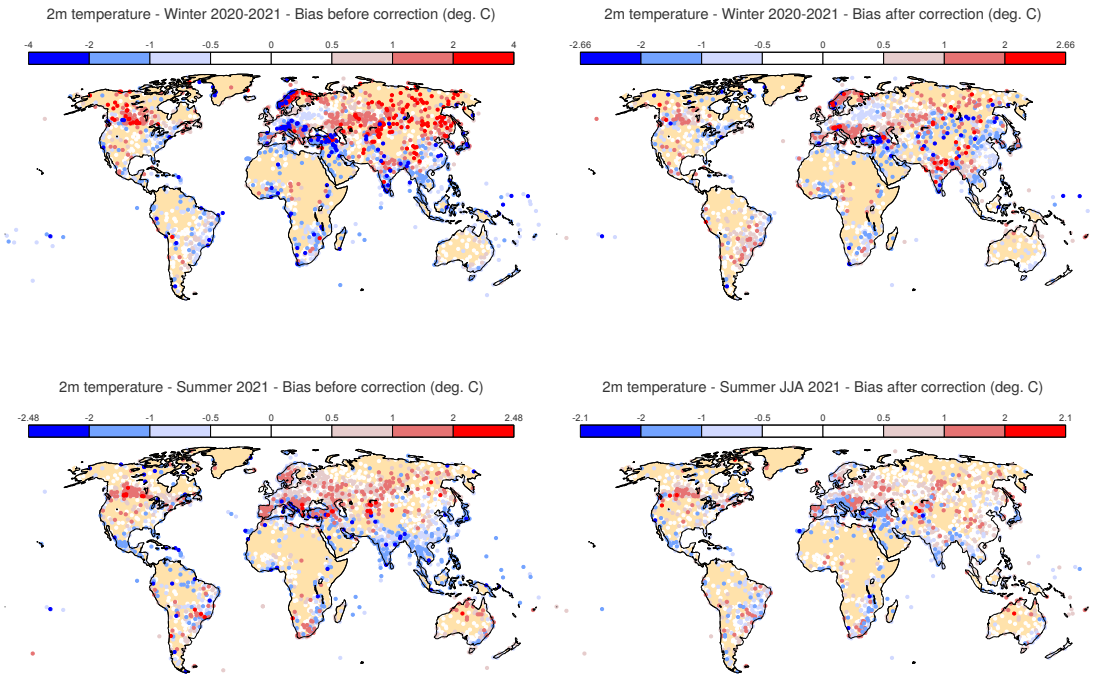

Figure 7: Forecast **bias** *of 2m temperature forecasts before (left) and after (right) correction using the combined model in winter (top) and summer (bottom). Results are aggregated over all lead times.*

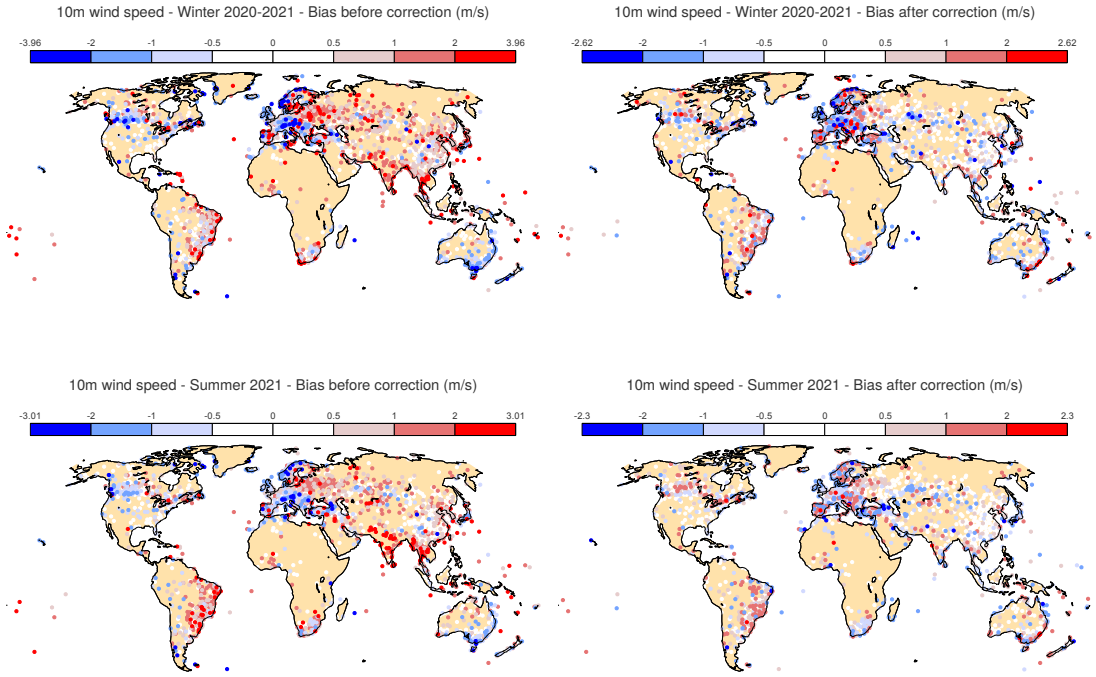

Figure 8: Forecast **bias** *as in Fig. 7 but for 10m wind speed.*

show here results for the combined predictions of the simpler *state-independent* configurations. Indeed, we only show results for the combined *state-independent* predictions as the combination improves the performance with respect to any *state-independent*-based models taken separately (not shown).

Global RMSE averages as a function of the forecast lead time are shown in Fig. 9. RMSE is reduced by around 10-15% for all lead times by all ML models with no self-imposed restrictions on the choice of predictors. The difference between ML models is much smaller than their difference to the default uncorrected forecast. For 2m temperature, the NN model performs slightly better than for the others and for 10m wind speed RF predictions are slightly better than the others. In all cases, the linear combination of models either slightly outperforms any single model or is extremely close. Changing the size and dates of the training data, quality control of the training data, and predictor selection appeared more important than the choice of the underlying ML method.

Building models using only static predictors and time indicators emerges as an appealing approach to postprocessing: a substantial share of the expected RMSE improvement with postprocessing is achieved with these simple and cost-effective model configurations. In general terms, there is a trade-off between complexity and applicability in an operational context on the one hand and postprocessed forecast performance on the other hand. For example, the combination of models leads to better results than individual models alone but at the cost of multiplying the models to be trained and maintained. Similarly, the combination of models of different types (as illustrated above) leads to better results than combining variants of the same model, as for example RF with different hyper-parameters (not shown).

Changes in RMSE are larger where RMSE errors are initially larger. There are great geographic variations in the RMSE of the default forecast for both 2m temperature and 10m wind speed forecasts (not shown). For example, the Alps are associated with an RMSE of around 4°C, while in northern France the

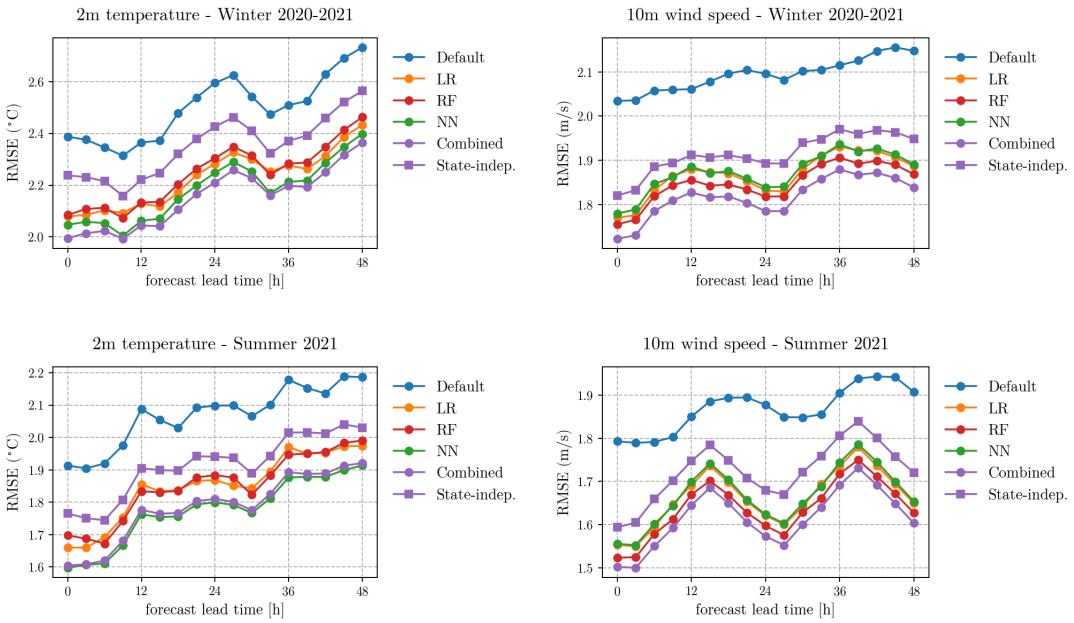

Figure 9: Forecast **RMSE** *as a function of the forecast lead time. Forecast of 2m temperature (left) and 10m wind speed (right), in winter (top) and in summer (bottom).*

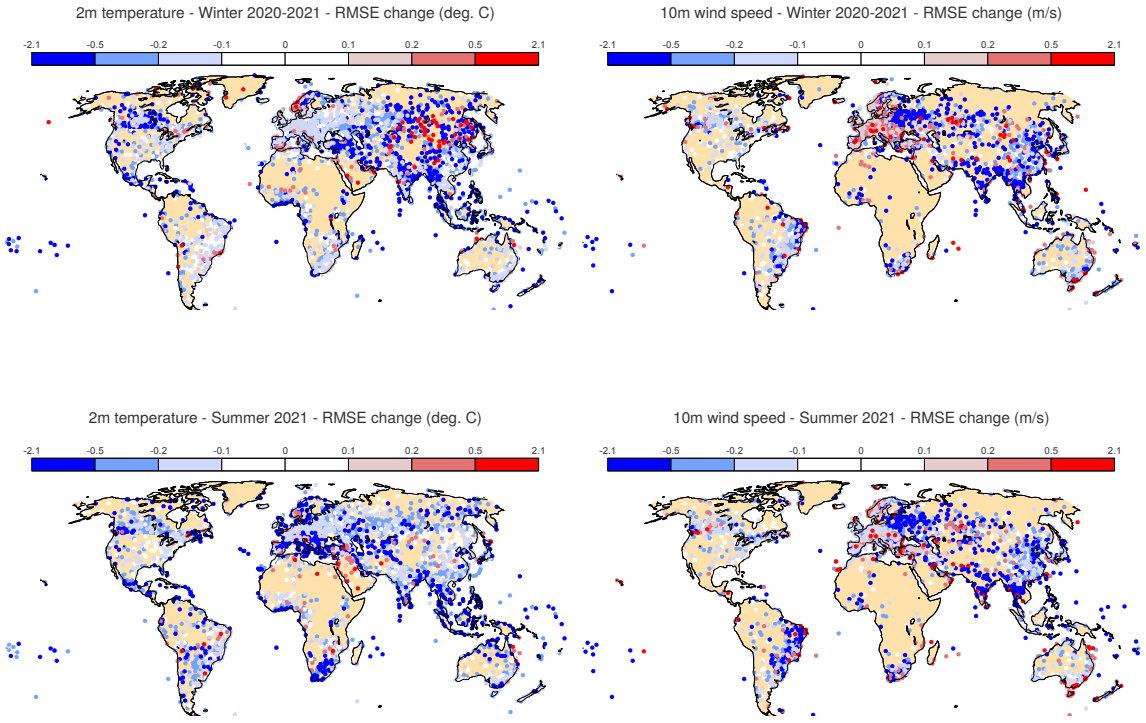

Figure 10: Change in performance in terms of **RMSE** *after bias correction. Blue colors indicate an improvement achieved with postprocessing. Forecasts of 2m temperature (left) and 10m wind speed (right), in winter (top) and in summer (bottom). Results are aggregated over all lead times.*

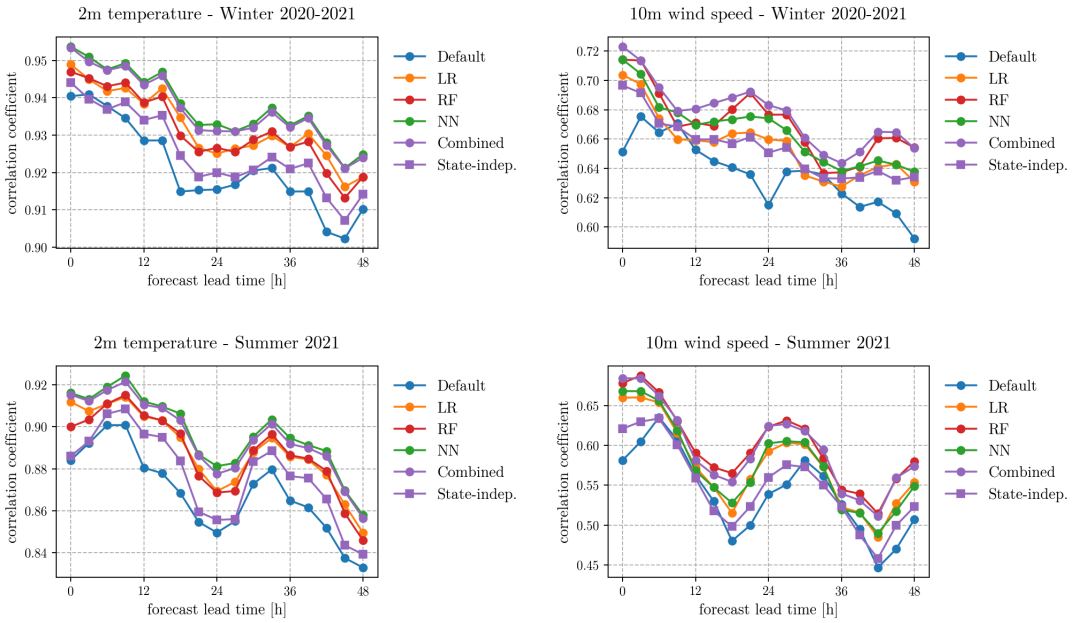

Figure 11: Same as Figure 9 but for the aggregated **correlation coefficients**.

RMSE is 1-1.5°C. The broad pattern of change is the same for all ML models, with the larger reductions in RMSE being in regions of high forecast RMSE. For both variables, RMSE has reduced in most parts of the world with some exceptions where all statistical models do not perform well as for example in East Asia for 2m temperature in winter or Central-Europe for 10m wind speed also in winter. We believe an increase in the size of the training dataset could help in such situations.

Finally, we also assess the strength of the linear relationship between forecasts and observations. At each station, the correlation coefficient between forecasts and observations is computed when observation measurements are available over the whole verification period (to avoid computing correlation coefficients on a small number of forecast/observation pairs). Coefficients are aggregated separately for each lead time in Fig. 11 and are consistent with RMSE results shown in Fig. 9. The correlation coefficient results also suggest that situation-dependent bias correction with ML techniques improves the ability of the forecast to capture the day-to-day weather variations.

5.3 Forecast uncertainty

Models for systematic and residual errors are developed sequentially (see Section 3). A second model (Model 2) focuses on the residual forecast error after bias correction. The resulting prediction is called *forecast uncertainty* and aims to reflect the level of confidence one can have in a forecast. On average, large (small) forecast uncertainty should be associated with large (small) forecast error. Statistical consistency between predicted forecast uncertainty and actual forecast error is called reliability and is checked with the help of reliability plots. The reader not familiar with these concepts could refer to Section 2.2 in Leutbecher and Palmer (2008).

In Figure 12, perfect reliability is indicated with a diagonal line. Results for all 3 types of ML approaches show good performance of the uncertainty models: overall the dots are close to the diagonal. Errors larger

than expected can occur when the forecast uncertainty is close to 0 (in particular for 2m temperature prediction in winter with NN and LR). We also see a general tendency of the NN models to under-predict forecast errors. The combined model provides in most cases a more reliable forecast.

The predicted forecast uncertainty could serve as a basis for delivering probabilistic forecasts. The first and second moments of an underlying forecast probability distribution can be derived from the systematic and residual error models, respectively. The uncertainty is valid at the point scale and so encompasses potential representativeness errors that cannot be captured by ensemble forecasting techniques. Also, as future work, the benefit of ML-based uncertainty models could be demonstrated using simple statistical models of representativeness errors as benchmark (as proposed recently in Ben Bouallègue *et al.*, 2020).

6 Conclusion

In this study, we performed statistical postprocessing of ECMWF's near-surface temperature and wind forecasts using 3 types of ML methods: linear regression, random forest, and neural network. After a rigorous selection of predictors, ML models are trained to predict situation-dependent bias and uncertainty of the high-resolution IFS global forecasts. Two distinct statistical models are used to infer systematic errors on the one hand and residual (random) errors on the other hand. This 2-model approach is applied to all 3 ML methods and feature importance analysis is performed for each error model individually. The source of random errors can therefore be explored independently of the source of systematic errors, with the first results indicating a close connection between the two types of error sources. The ML-based statistical models allow delivering postprocessed forecasts not only at locations of observation measurements but also at any other points on the globe. The promising results obtained for deterministic forecasts at short lead times encourage further research involving the ECMWF ensemble forecasts as well as longer lead times. In addition, the discussed ML approaches would be easily transferable to other weather variables such as precipitation.

Weather-dependent bias correction with ML techniques notably improves the forecast, with a reduction between 10 and 15% in terms of RMSE for all lead times and variables envisaged here. In essence, our study shows that the accuracy of the postprocessed forecasts does not depend so much on the choice of the ML method but more crucially on the selection of predictors, the size of the training and test datasets, and the quality control applied to the data. In this context, we have identified ML-based solutions for forecast postprocessing with different levels of complexity in terms of practical implementation. *state-independent* postprocessing configurations that only rely on predictors available before the start of the forecast are simple to implement and easy to maintain. Reduction in forecast error can be further improved with the help of more complex configurations involving state-dependent predictors and/or the combinations of ML models. Finally, the good performance of the forecast uncertainty models opens new horizons for the generation of calibrated probabilistic weather forecasts based on statistical models.

Acknowledgements

For this work, Fenwick Cooper was partially funded by iFAB (International Foundation Big Data and Artificial Intelligence for Human Development, www.ifabfoundation.org). Matthew Chantry gratefully acknowledges funding from the MAELSTROM EuroHPC-JU project (JU) under No 955513.

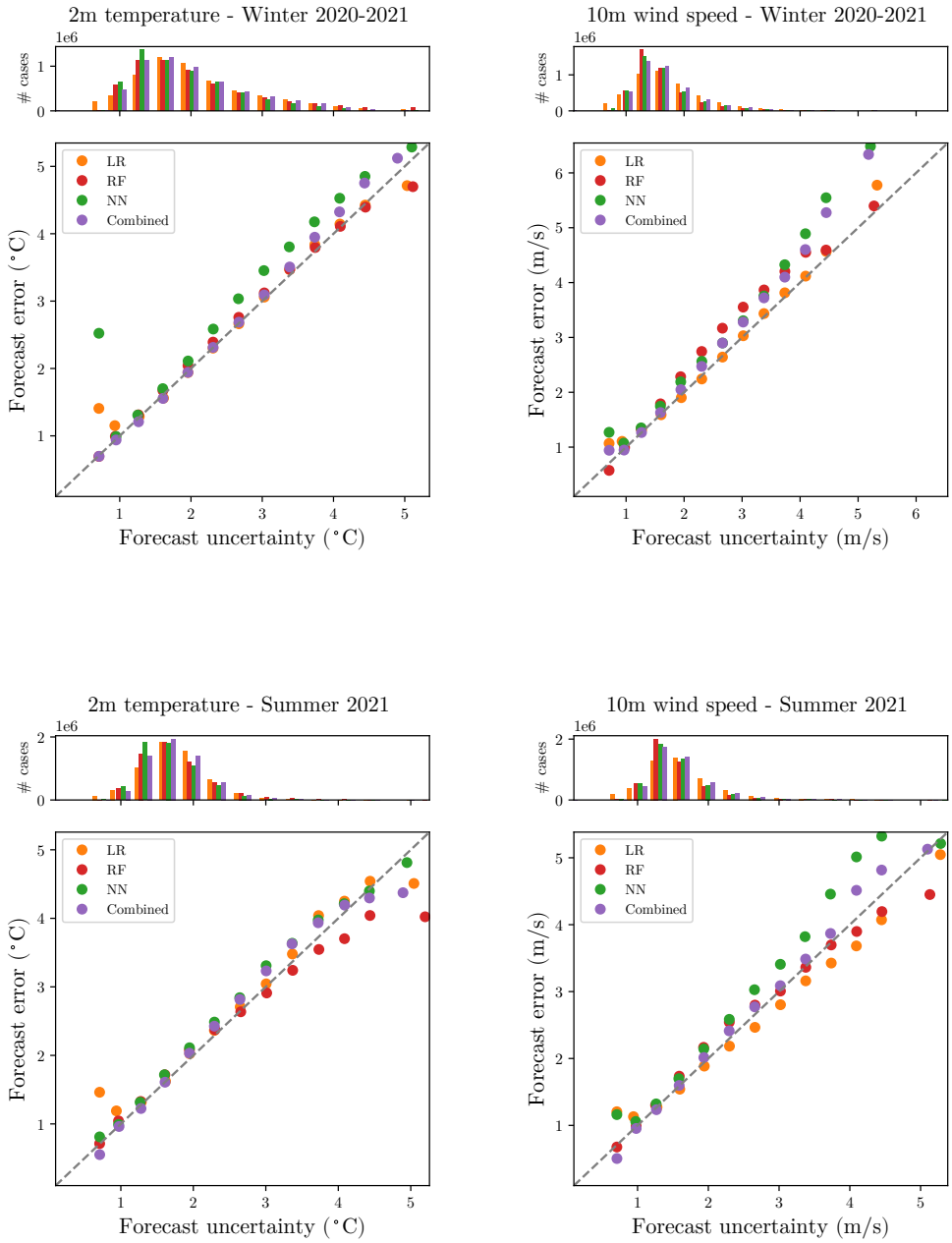

Figure 12: ***Reliability*** *plots showing the uncertainty/error relationship for 2m temperature (left) and 10m wind speed (right) postprocessed forecasts, verifying over summer (top) and winter (bottom). The forecast uncertainty (x-axis) corresponds to the square root of the residual error prediction, the actual forecast error (y-axis) corresponds to the RMSE of the bias-corrected forecast. Perfect reliability is indicated with a dashed diagonal line. For each plot, a histogram shows the number of cases in each forecast uncertainty category.*

References

Ben Bouallègue, Z. (2017). Statistical postprocessing of ensemble global radiation forecasts with penalized quantile regression. *Meteorologische Zeitschrift*, **26**(3), 253–264, doi:10.1127/metz/2016/0748.

Ben Bouallègue, Z., Haiden, T., Weber, N. J., Hamill, T. M. and Richardson, D. S. (2020). Accounting for representativeness in the verification of ensemble precipitation forecasts. *Monthly Weather Review*, **148**(5), 2049–2062, doi:10.1175/MWR-D-19-0323.1.

Düben, P., Modigliani, U., Geer, A., Siemen, S., Pappenberger, F., Bauer, P., Brown, A., Palkovic, M., Raoult, B., Wedi, N. and Baousis, V. (2021). Machine learning at ecmwf: A roadmap for the next 10 years. *ECMWF Technical Memorandum*, **878**.

ECMWF (2020). *IFS Documentation CY47R1 - Part III: Dynamics and Numerical Procedures*. doi:10.21957/u8ssd58, URL https://www.ecmwf.int/node/19747.

Haiden, T., Janousek, M., Vitart, F., Ben Bouallègue, Z., Ferranti, L. and Prates, F. (2021). Evaluation of ECMWF forecasts, includuing 2021 upgrade. *ECMWF Technical Memorandum*, **884**.

Hamill, T. M. (2021). Comparing and combining deterministic surface temperature postprocessing methods over the united states. *Monthly Weather Review*, **149**(10), 3289 – 3298, doi:10.1175/MWR-D-21-0027.1.

Hemri, S., Scheuerer, M., Pappenberger, F., Bogner, K. and Haiden, T. (2014). Trends in natural calibration of raw ensemble weather forecasts. *Geophys. Res. Lett.*, **41**, 9197–9205, doi:10.1002/2014GL062472.

Leutbecher, M. and Palmer, T. (2008). Ensemble forecasting. *Journal of Computational Physics*, **227**(7), 3515–3539, doi:https://doi.org/10.1016/j.jcp.2007.02.014, predicting weather, climate and extreme events.

Louppe, G., Wehenkel, L., Sutera, A. and Geurts, P. (2013). Understanding variable importances in forests of randomized trees. *Advances in Neural Information Processing Systems*, pp. 431–439.

McGovern, A., Lagerquist, R., John Gagne, D., Jergensen, G. E., Elmore, K. L., Homeyer, C. R. and Smith, T. (2019). Making the black box more transparent: Understanding the physical implications of machine learning. *Bulletin of the American Meteorological Society*, **100**(11), 2175–2199, doi:10.1175/BAMS-D-18-0195.1.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (2007). *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, third edition.

Rasp, S. and Lerch, S. (2018). Neural networks for post-processing ensemble weather forecasts. *Mon. Weather Rev.*, **146**(11), 3885–3900, doi:10.1175/MWR-D-18-0187.1.

Sandu, I., Haiden, T., Balsamo, G., Schmederer, P., Arduini, G., Day, J., Beljaars, A., Ben-Bouallegue, Z., Boussetta, S., Leutbecher, M., Magnusson, L. and de Rosnay, P. (2020). Addressing near-surface forecast biases: outcomes of the ecmwf project 'understanding uncertainties in surface atmosphere exchange' (usurf). *ECMWF Technical Memorandum*, **875**.

Taillardat, M., Mestre, O., Zamo, M. and Naveau, P. (2016). Calibrated ensemble forecasts using quantile regression forests and ensemble model output statistics. *Mon. Weather Rev.*, **144**(6), 2375–2393, doi: 10.1175/MWR-D-15-0260.1.

Vannitsem, S., Bremnes, J. B., Demaeyer, J., Evans, G. R., Flowerdew, J., Hemri, S., Lerch, S., Roberts, N., Theis, S., Atencia, A., Ben Bouallègue, Z., Bhend, J., Dabernig, M., Cruz, L. D., Hieta, L., Mestre, O., Moret, L., Plenkovic, I. O., Schmeits, M., Taillardat, M., den Bergh, J. V., Schaeybroeck, B. V., Whan, K. and Ylhaisi, J. (2021). Statistical postprocessing for weather forecasts: Review, challenges, and avenues in a big data world. *Bulletin of the American Meteorological Society*, **102**(3), E681 – E699, doi:10.1175/BAMS-D-19-0308.1.

WMO (2019). Manual on the global data-processing and forecasting system. *Annex IV to the WMO Technical Regulations*, appendix 2.1.2.