Predicting Weather Forecast Uncertainty with Machine Learning

Sebastian Scher[1] | **Gabriele Messori**[1]

[1]Department of Meteorology and Bolin Centre for Climate Research, Stockholm University, Stockholm, Sweden

Correspondence
Sebastian Scher, Department of Meteorology and Bolin Centre for Climate Research, Stockholm University, Stockholm, Sweden
Email: sebastian.scher@misu.su.se

Funding information
Department of Meteorology of Stockholm University, Vetenskapsrådet grant no. 2016-03724

Weather forecasts are inherently uncertain. Therefore, for many applications forecasts are only considered valuable if an uncertainty estimate can be assigned to them. Currently, the best method to provide a confidence estimate for individual forecasts is to produce an ensemble of numerical weather simulations, which is computationally very expensive. Here, we assess whether machine learning techniques can provide an alternative approach to predict the uncertainty of a weather forecast given the large-scale atmospheric state at initialisation. We propose a method based on deep learning with artificial convolutional neural networks that is trained on past weather forecasts. Given a new weather situation, it assigns a scalar value of confidence to medium range forecasts initialised from said atmospheric state, indicating whether the predictability is higher or lower than usual for the time of the year. While our method has a lower skill than ensemble weather forecast models in predicting forecast uncertainty, it is computationally very efficient and outperforms a range of alternative methods that do not involve performing numerical forecasts. This shows that it is possible to use machine learning in order to estimate future forecast uncertainty from past forecasts. The main constraint in the performance of our method seems to be the number of past forecasts available for training the machine learning algorithm.

KEYWORDS
Machine learning, Statistical methods, Weather forecast, Ensembles

[*] S. Scher has designed the study, developed and implemented the machine learning methods, analyzed the results and prepared the manuscript. G. Messori prepared the d and θ data and contributed to analyzing the results and drafting the manuscript.

1 | INTRODUCTION

Weather forecasts are a major benefit for society and sustainable development (UNISDR, 2007). Over the last decades, numerical weather prediction (NWP) models have been improving steadily, providing more and more accurate forecasts (Bauer et al., 2015). However, the atmosphere is a highly chaotic system. Therefore, perfect forecasts are impossible, and every weather forecast is, to a certain degree, uncertain (Lorenz, 1963). It has long been recognized that, for many applications, a weather forecast is only valuable if some measure of confidence can be attached to the forecast. Ideally, one would create some measure that considers the state of the atmosphere the forecast is initialised from as input, and then judges the extent to which it may lead to a good or bad forecast. Ferranti et al. (2015) showed that in certain large scale atmospheric flow regimes, predictability in the medium range (∼3-14 days) is higher than in other regimes. However, the variability of predictability within one flow regime is typically higher than the differences in the mean predictability between the flow regimes. Therefore, forecasting predictability on a day-to-day basis remains a non-trivial task. This problem has been alleviated by the introduction of ensemble forecasts in the 1990s. Instead of producing a single forecast, a weather forecast model is run multiple times with slightly different initial conditions and/or slightly different model formulations or stochastic parameterizations. This results in an ensemble of forecasts being produced (see for example Gneiting and Raftery (2005)). If all (or at least most of) the ensemble members show good agreement - that is, they have a low spread and thus produce very similar forecasts - the weather situation is highly predictable, and the confidence in the forecast high. If they show very different forecasts (high spread), the current weather situation is less predictable and confidence in the forecast is low. While once in a while a NWP model can produce a very accurate forecast in an inherently unpredictable situation (i.e. a situation with large forecast spread), on average the forecast error correlates with the forecast uncertainty (and spread). Ensemble forecasts have gained widespread acceptance in the meteorological community and are routinely used by major forecasting centers around the world. Their major drawback is, however, that they are very computationally expensive to produce. Due to this, they also often have a time-lag of several hours between initialisation and delivery to end-users, even when run on a super-computer. Here, we try to assess an alternative method to assign a confidence measure to weather forecasts. It is exclusively based on past forecasts, and does not require running a NWP model. This approach could therefore be used as a complement to ensemble forecast systems. Our method is based on the hypothesis that the forecast uncertainty is strongly dependent on the inherent predictability imposed by the atmospheric state it is initialized from, plus a random component. Therefore, if a long enough series of past forecasts and their spread/error are available, it should be possible to infer which situations are likely to lead to higher and which situations are likely to lead to lower forecast uncertainty. The connection between initial atmospheric fields and the forecast spread/error is, however non-trivial and non-linear, and no effective methods to diagnose it have been proposed up to now. However, recent advances both in the availability of reforecast datasets and machine learning techniques may now allow to exploit this link.

Artificial neural networks have been used in meteorology and climate science for a couple of decades. They have been applied succesfully for finding empirical relationships between meteorological variables (Hewitson and Crane, 1992) as well as in climate attribution studies (Pasini et al., 2017) and for El-Nino forecasting (e.g. Hsieh and Tang (1998); Tangang et al. (1998). A good overview of early applications can be found in (Hsieh and Tang, 1998). Additionally, Neural networks and the closely related support vector machines have been effective in downscaling global climate models (e.g. Dibike and Coulibaly (2006); Ghosh and Mujumdar (2008)), where they are used to map large-scale atmospheric states to regional weather. Other applications include, for example, cloud classification (Bankert, 1994), tornado forecasting and detection (Marzban and Stumpf, 1996), radar quality control (Lakshmanan et al., 2014; Anagnostou, 2004) and real-time decision-making in high-impact weather situations (McGovern et al., 2017). Recently, deep convolution networks - a subclass of artificial neural networks - have shown very high skill in various non-linear problems such as image recognition (Schmidhuber, 2015), and were also applied to detecting extreme weather in climate datasets (Liu et al., 2016).

Here, we present two closely related methods that use a Convolutional Neural Network (CNN) to link atmospheric states to

forecast uncertainty. Both methods take as input the initial analysis of a NWP model (the current state of the atmosphere in the model representation), and output a measure of uncertainty of the forecast at a certain lead time within the medium range. The first method is trained on the past errors of a deterministic weather forecast model, while the second method is trained on past spread of an ensemble weather forecast model. The first method is more general, because it only requires past forecasts from a deterministic model. The second method should be easier to train, but is less generally applicable because it needs past ensemble forecasts, which are less available.

To aid the reader, we provide here a summary of the terminology in this paper: the terms "forecast" and "prediction" are often used interchangeably in the meteorological literature. "Prediction", however, is also widely used in statistics and machine learning, with a different meaning. In order to avoid confusion, we will always refer to weather forecasts as "forecasts", and not as "predictions" (except when we use the standard terminology: "numerical weather prediction model"). "Prediction" will be used in order to refer to the uncertainty that our methods assign to a forecast. The (inverse) uncertainty of the forecast is also often called the "predictability" of the atmosphere. Note that "predictability" and "prediction" are used in different ways: our methods predicts the predictability.

2 | DATA

As reference forecast data we use the second version of the GEFS reforecast dataset (Hamill et al., 2013). Operational NWP models are updated regularly (roughly every 6 months), and thus cannot provide a continuous set of homogeneous forecasts. Reforecasts mitigate this problem. They are forecasts that use observations from the past in combination with a single version of a state-of-the-art NWP model, thus producing a long series of relatively homogeneous forecasts. The use of the term "relatively" is dictated by changes in the observation system, which may introduce inhomogeneities. Still, reforecasts are currently the best available long-term forecast timeseries. The GEFS reforecasts provide 16-day forecasts initialized daily at 00:00 UTC from Dec 1984 to present. They are updated operationally every day. The forecasts consist of 10 members and one unperturbed control forecast. The resolution of the model is T254L42 ($\sim$40km) for the first 8 forecast days. We chose the GEFS reforecast dataset because it is currently the longest publicly available set of daily reforecasts.

When choosing input variables for the machine learning methods, a trade-off between completeness, data size and avoidance of noise has to be made. Using all available data from the initial model analysis would provide the most information, but potentially also include a lot of information and noise not necessary to address our specific problem. This would make the training process of the machine-learning algorithm more difficult due to the larger amount of data. Therefore, we limit our training set to three variables: geopotential height at 500hPa and meridional and zonal wind at 300hPa. These provide a good overview of instantaneous atmospheric states and their predictability in the medium range. Tests using the same variables on additional pressure levels showed no clear improvement in our method's skill (not shown). The choice of the region for the input analysis is based on the same trade-off between data size and completeness. We focus on the region 165°W - 60°E, 0°N-80°N (see fig. 1). This was chosen due to the fact that our target region is Europe (see below), and in the mid-latitudes there is typically a downstream dependence of forecast errors (e.g. Simmons and Hoskins (1979)). Thus, the domain extends further west than east relative to the target region. Furthermore, it has been shown that the source of forecasts with exceptionally high errors above Europe is in the Atlantic sector and North America (Rodwell et al., 2013). Tests using smaller domains (down to 165°W - 0°E) showed only a modest decrease in our method's skill (not shown).

As a baseline measure of atmospheric predictability, we use the spread of the GEFS reforecasts, computed as the ensemble standard deviation of the 500hPa geopotential height of the forecast 3-6 days after initialisation. The spread in 500hPa geopotential height is a good proxy for the predictability of the large scale weather situation, as it is highly correlated with surface temperature spread (not shown), and additionally also indicates the uncertainty in predicting cyclones. Therefore, in our study we assume

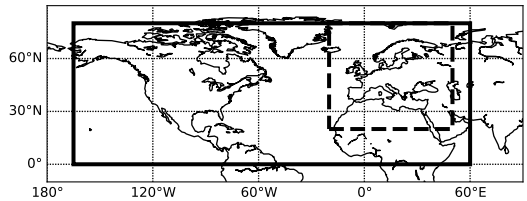


FIGURE 1 Domain of atmospheric input fields (solid box) and domain over which spread and error are computed (dashed box).

that it is the best single proxy for the real medium-range predictability. The spread is computed for a section encompassing the North Atlantic and Europe (-20°E to 50°E 20°N to 80°N, see fig. 1). This region is widely used in assessing the forecast skill of medium range forecasts (e.g. Ferranti et al. (2015)). Additionally, we compute the root mean square error of the 500hPa geopotential height of the control forecast for the same region, for forecast days 3-6. This measure will be referred to as the forecast error.

Each of the input variables and the target values (RMSE and spread) are separately normalized to zero mean and unit variance. The forecast error shows a small positive trend with time, likely due to the increase in available observations (Hamill et al., 2013). Therefore we (linearly) detrend the forecast error. We note that the spread does not show such a trend, and we therefore do not detrend it.

The period 1985-2016 contains approximately 10.000 days. Since we have daily forecasts, this means that we have a dataset of approximately 10.000 samples. The data was split up into a training set, a validation set and a test set. The training set is the principal dataset on which the neural network is trained. The validation set is used for tuning the algorithm's hyper-parameters (see methods section), namely the architecture of the network. It is also used to control over-fitting (the situation where the trained network predicts the target uncertainty within the training period with very high skill, but with very low or no skill in the test period). The test set is neither used for training nor for tuning, but is instead used for validation of the results. All results in this paper are reported for the test dataset.

For machine-learning tasks, data is often split up randomly in train, test and validation sets. However, both our input data (atmospheric fields) and target values (spread and error) have non-zero autocorrelation. Therefore, random splitting is inappropriate. If, for example, for a specific test case both the day before and afterwards are in the training data, it would be much easier for any learning algorithm to predict the target value for the test case. Therefore, we split our data in the following way: the years 1990 and 2008 are used for validation, the years 2010-2016 for testing, and all other years for training. This is inspired by an operational context. In this case, past forecasts and the corresponding spreads/errors are available, and predictions are to be made for future fields. The choice of 6 years in the test data was made to get representative and significant results while retaining a sufficiently long training set. The choice of using 2 separate years in the validation data was made in order to account for potential low-frequency effects.

Both forecast spread and forecast error have a very pronounced seasonality in mid-latitudes. On average, spread and error are lower in summer than in winter. Therefore, the climatology already provides seemingly high skill in predicting the absolute spread of a given ensemble forecast (correlation ∼0.71, not shown). This seasonality is of the same order or larger than day-to-day anomalies with respect to the seasonal cycle. However, the day-to-day variability in predictability is of much higher interest. Therefore, we focus our study on predicting spread and error *anomalies*. The anomalies are computed via subtracting a 30-day running climatology. Thus, for 15th January 2000, the mean of all January days from 1985-2016 is subtracted. This very effectively removes the seasonality: the correlation between climatology and anomalies is reduced to 0.02 for spread and 0.01 for error. Hereafter, "error" and "spread" will always refer to error and spread *anomalies*. Note that because of the high skill of

climatology in predicting absolute spread, when predicting the anomalies, the reported skill will usually be lower than when predicting absolute spread. Therefore, when comparing our results to other methods, this should be taken into account.

3 | METHODS

3.1 | Convolutional Neural Networks

Artificial Neural networks are in principle a series of non-linear functions applied to input data, finally outputting one or more variables. Convolutional Neural Networks (CNNs) are a sub-group of artificial neural networks often used in deep learning. They made their breakthrough in 2012 (Krizhevsky et al., 2012). Since then, they have been the state-of-the-art in image recognition and various other fields (LeCun et al., 2015). When representing meteorological data on regular 2-D grids, typical CNN setups from image recognition tasks can be directly applied to meteorological data. In contrast to other methods, like multi-linear regression, there is no need for dimensionality reduction. For a detailed description of deep learning and CNNs, we refer the reader to LeCun et al. (2015). At training time, we present our network with the input atmospheric fields and the target values (error or spread). At testing time, only the input fields are provided, and the network predicts the output value - the predicted forecast uncertainty.

While the parameters of a neural net are determined during training, the choice of the network architecture has to be made beforehand. As there is no known procedure to define the best architecture beforehand for a new type of problem, this is usually done on a trial-and-error basis. To the best of the authors' knowledge, no architecture explicitly designed for the problem analyzed in this study has yet been proposed. Therefore, we chose a principal network architecture that is commonly used in image recognition tasks, and then tried out a large number of different configurations of the network, finally using the one that worked best on the validation data. The principal architecture is: 2 convolution layers, followed by a max-pooling layer, followed by 2 more convolution layers and a max-pooling layer, followed by a fully connected hidden layer and finally a single linear output neuron. All neurons except the linear output neuron use the Relu (rectified linear unit) activation function. After each max-pooling layer and the fully connected layer a drop-out layer is added. The max-pooling layers have stride 2x2. In the exploratory phase, we varied the size of the hidden layer, the type of convolution layers (normal convolution layer, convolution layer followed by a batchnormalization layer, residual layer), the depth of the convolution layers and the dropout-probability in the drop-out layers. In addition to the network architecture, the learning rate parameter has to be specified before training. The best value of this parameter was also determined in the trying-out-phase. Additionally, we also tried out simpler machine learning methods (Support Vector Regression, principal component analysis followed by linear regression, principal component analysis followed by random forest error regression). All of them were clearly outperformed by the CNN method, and they will therefore not be described in this study.

Our final network choice has normal convolution layers, a hidden layer size of 32, a convolution depth of 32, and a dropout-probability of 0.7. The three input variables are represented as separate channels in the input layer. The training is done using the Adam-optimizer (Kingma and Ba, 2014) with root mean square error as loss function (root mean square error of the output variable, not to be confused with the root mean square error of the model forecast). The network is trained for 30 epochs (iterations over the dataset), and then the epoch on which the skill on the validation dataset is highest is selected.

After the training, an additional rescaling is added to increase the prediction skill in case there is a large bias in the predictions: the predicted values are rescaled such that the predicted values for the training data have the same mean and variance as the target values for the training data. The predicted values for the validation and test dataset are then rescaled with the same scaling factors as the training data. This rescaling is a statistical post-processing, using exclusively the training data. The neural network is built with the open-source libraries Keras (Chollet et al., 2015) and the TensorFlow backend (Abadi et al., 2015).

In the training of the network, there are various random components. Therefore, we train the network 10 times, in order to

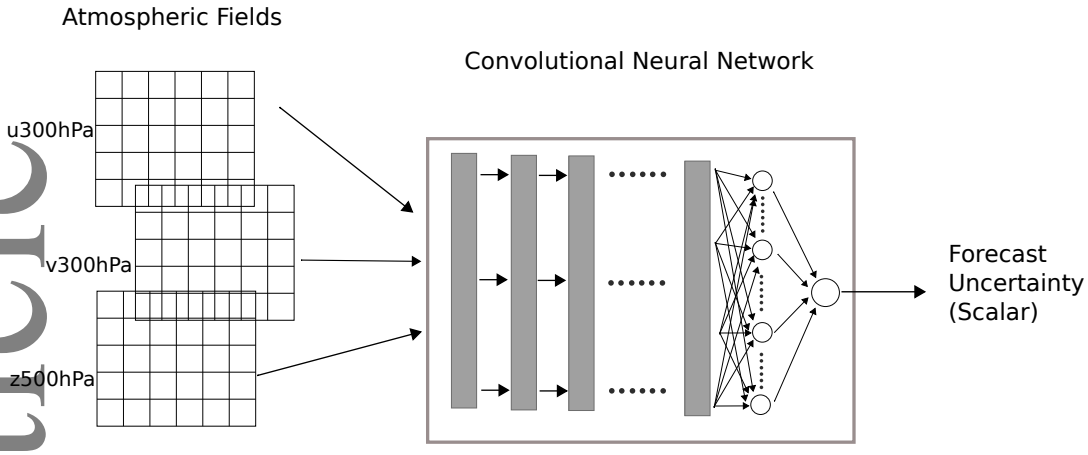


FIGURE 2 Sketch of the implemented method. Left the input data (2D fields), in the center the convolutional neural network, and to the right the output (one scalar value representing predictability).

make sure that our results do not occur by chance. In a number of cases, the network completely fails to learn. Therefore, we do a post-selection of the network ensemble. Only networks that have at least a correlation of 0.15 between the prediction and the target values for the training set are retained. With this post-selection usually 5-10 members are retained from one ensemble. Note that as we only use the performance on the training set as selection criterion, this is neither a tuning on the validation nor on the test data. As final prediction, we use the mean of the predictions of the sub-selected network ensemble. Training 10 members takes around 30 minutes on 2 NVidia K80 GPUs. An overview of our approach is sketched in fig. 2.

3.2 | Baseline Methods

In order to evaluate our approach, we compare it to two methods that have been proposed in the literature - namely Persistence/Local Dimension, and weather type clustering. Additionally, we compare it to a simple nearest neighbor approach. Persistence/local dimension have recently been proposed for use in operational weather forecasting, but this is yet to happen. Weather type clustering has been proposed mainly for analytic purposes (to study under which conditions weather forecasts are more and less skillful) and not for operational use. The nearest neighbor method has not been proposed for operational practice, but is included here as a reference "minimal" machine learning method.

| Persistence and Local Dimension

Faranda et al. (2017) recently proposed a method that computes the inverse persistence in phase space (θ) and the local dimension (d) of surface pressure fields. They propose that d and θ can be used as indicators for atmospheric predictability in the medium range. The original analysis by Faranda et al. was done on ERA-Interim and NCEP/NCAR-reanalysis data. For a fair comparison here, the two indicators are computed for the analysis fields of the GEFS-reforecasts, for the same input region used for the machine-learning approach (see fig. 1). In contrast to both our machine learning approach and the other baselines, d and θ have the advantage that they do not rely on past forecasts, but only need past analysis fields. For the exact definition and computation of d and θ we refer the reader to Faranda et al. (2017).

| Weather Type clustering

Ferranti et al. (2015) showed that both the error and the spread of medium range forecasts above Europe is dependent on the initial weather type in the Euro-Atlantic sector. They used 4 weather types, defined by k-means clustering of the first 10 EOFs of the 500hPa geopotential height field. These weather types correspond to the well known NAO+, NAO-, Atlantic Ridge and Blocking flows. This raises the question of whether the current weather type can be used as an indicator of predictability. For this, we compute the mean spread for all 4 weather types in the training data. For each test day, the mean spread corresponding to the current weather type is used as prediction.

| Nearest Neighbor

One of the simplest methods of "learning" from past data is to simply look for cases in the past that were similar to the current situation. For the nearest-neighbor baseline, for every day in the test dataset we look for the day with most similar atmospheric fields in the training dataset, based on an L2 norm. The spread of the forecast from the selected day is then used as prediction of the uncertainty for the test day.

4 | RESULTS

In this section, we present the results of our machine learning approach. We start by comparing the predicted and actual forecast model spread. We then assess how well our predictions do in differentiating between days with high and low forecast errors. In section 4.3, we discuss how dependent the skill of our method is on the length of the training data set. Finally, we compare our method to the baseline methods described in section 3.2.

4.1 | Comparison with Model Spread

When trained on past ensemble spread, the correlation between the predicted uncertainty of our network and the real ensemble spread is ∼0.33 for forecast day 3 (fig. 3 left panel). The more difficult problem of training on model error has slightly lower correlation with the model spread (∼0.27). The skill decreases with increasing lead-time, down to ∼0.28 (trained on spread) and 0.19 (trained on error) for forecast day 6 (not shown). Even though the correlations are not very high in absolute terms, they are highly significant (see p-values in fig. 3). This means that, on average, the forecast spread is high when the predicted uncertainty is high, and the forecast spread is low when the predicted uncertainty is low. In order to account for potential low-frequency effects, we repeated our analysis using the beginning of the available reforecast period (1985-1991) as test dataset and training the algorithm with the rest of the dataset. The skill we obtained is slightly lower, with a correlation of 0.24 for forecast day 3 when trained on spread (not shown), but still highly significant.

Splitting up the test data according to seasons (thus only analyzing all winter months, summer months etc) reveals that there is some seasonal dependence on the skill of our method in predicting uncertainty, especially at longer lead times (fig. 4). For forecast day 3, MAM, JJA and SON have roughly the same skill (correlation of ∼0.30), but in DJF the correlation is higher (∼0.40). For forecast day 6, skill in summer is below 0.10, and in SON around 0.15, but for wintertime and spring is above 0.30. Thus even at later forecast days, there is still a highly significant correlation, at least in spring and wintertime (see uncertainty-bars in fig. 4). The fact that also the training on past errors leads to significant correlations with the model spread is a very encouraging result given the difficulty of the problem: the goal is to get a measure of predictability. Past forecast errors are however - as mentioned earlier - only on average a measure of predictability. Therefore, when training on past error, the network is presented with a considerable number of "wrong" training cases (low error even though predictability is low). It is therefore

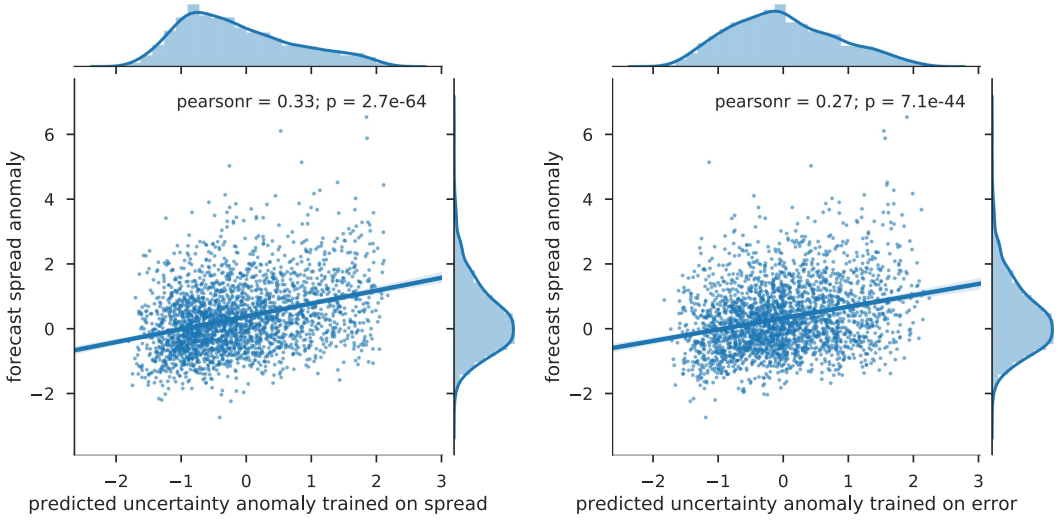


FIGURE 3 Predicted forecast uncertainty vs ensemble spread anomaly, trained on past ensemble spread (left) and past error (right), for forecast day 3. The shading around the regression line shows the uncertainty (5-95) estimated via bootstrapping.

encouraging that the network is still able to generalize from this sub-optimal training data.

Our main interest lies in predicting the day-to-day anomalies of forecasts (e.g. whether today's forecast is more uncertain than yesterday's forecast), which can best be achieved via looking at anomalies with respect to climatology. Still, it is interesting to also assess what our predictions can add to the climatology. Therefore, we turn our attention not to the forecast anomalies, but to the full uncertainties (forecast spread climatology plus anomaly). For this, we compute the fraction of days where the spread climatology plus the predicted spread are closer to the real spread than the climatology. The result is shown in fig. 5. As can be seen, for all lead days, the predicted full uncertainty beats the climatology in 60-62% of the cases for the network trained on error, and in 64-66 % for the network trained on spread.

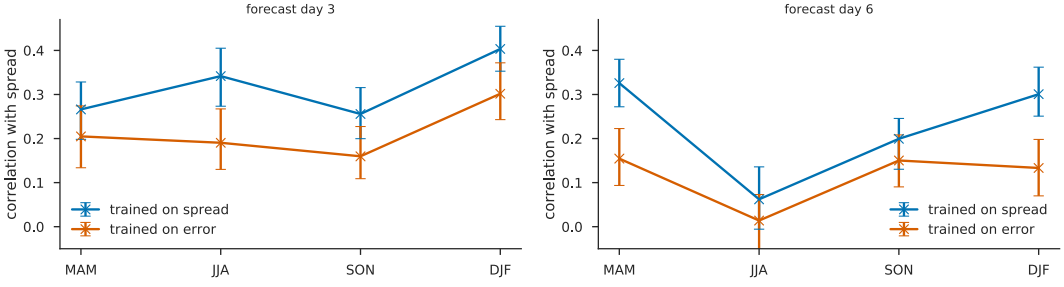


FIGURE 4 Seasonal dependence of the correlation between predicted forecast uncertainty and ensemble spread anomaly, for the networks trained on past forecast spread (blue) and error (red), at day 3 (left) and day 6 (right). The error bars show the uncertainty (5-95) estimated via bootstrapping.

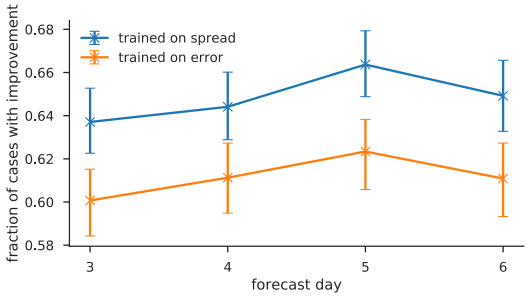


FIGURE 5 Fraction of cases where spread climatology plus predicted forecast uncertainty is closer to the real ensemble spread than climatology alone. The error bars show the uncertainty (5-95) estimated via bootstrapping.

4.2 | Comparison with Forecast Error

Correlating our uncertainty predictions with the forecasts' ensemble spread provides an initial evaluation of the effectiveness of our approach. However, it is non-trivial to define a usability threshold in terms of correlation (i.e. "uncertainty predictions with correlation of X or higher with the model spread are useful"). Therefore, we proceed to assess how well our uncertainty predictions do in differentiating between days with high and days with low forecast error.

When evaluating numerical ensemble weather forecasts, it is standard practice to look at the so-called spread-skill relationship. This shows how well the spread (the forecast uncertainty as given by the ensemble model) relates to actual skill (error) of the forecasts. Due to the already mentioned imperfect correlation between predictability and forecast error (see introduction), simple scatterplots of spread vs error are hard to interpret and can be misleading. Therefore it is common to bin the forecasts according to their spread. For each bin, the mean error of the forecasts in the bin is computed and plotted. Such plots are for example routinely shown for ECMWF's operational weather model on the ECMWF's website (ECMWF, 2018). We use this approach, but in our case with the predictions of our network rather than with the spread of the forecast model. The results are shown in fig.6. Panel a) shows the predicted uncertainty vs error relation for the network trained on spread, panel b) for the network trained on error. Additionally to the mean error in each bin, the 40-60 and 20-80 percentile ranges are shown. As reference, we also include the spread skill relationship of the forecast model, for the same time-period (panel c). The binning was done such that each bin contains 300 data points. Using a smaller number of data points per bin leaves the overall shape similar, but makes the picture noisier, especially in the high and low percentiles.

The most important feature of a good indicator of forecast uncertainty is that the binned forecast error means increase monotonically with increased predicted uncertainty. This is roughly the case for the network trained on spread, except for a small dip in the middle, and a leveling off at the end with high predicted uncertainty. For the network trained on error, the relation is similar. This is in agreement with the earlier result that both training methods have somewhat comparable performances in predicting forecast uncertainty. Therefore, both methods seem to be useful indicators of forecast uncertainty.

In order to assess the discriminative power of our uncertainty prediction system more quantitatively, we compute the F1 score with respect to classifying forecasts as either being below or above average error. The F1 score is defined as the harmonic mean of precision (PR, often known positive predictive value) and recall (RC, often known as hit rate):

$$F1 = 2\frac{PR \cdot RC}{PR + RC}$$

It ranges from 0 to 1, with 1 indicating a perfect classification. In the special case of forecast error, however, even a perfect predictor would have a score lower than 1, again due to the fact that the spread-error correlation is not 1 (see above). In order to

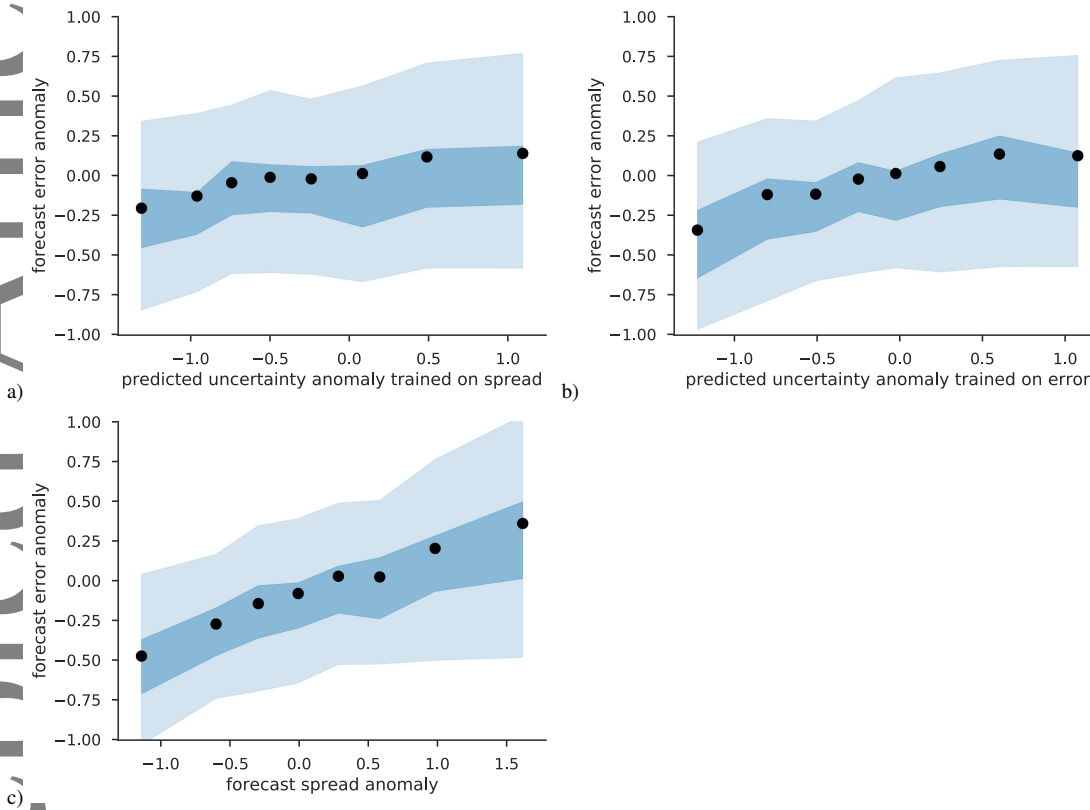


FIGURE 6 Relationship between predicted uncertainty and actual forecast error. The data was binned along the x-axis in bins containing 300 samples. The points show the mean forecast error of one bin. The shading shows the 20-80 and 40-60 percentile range of forecast errors for each bin. a) neural network trained on spread, b) neural network trained on error, c) spread of the forecast model. All panels refer to forecast day 3.

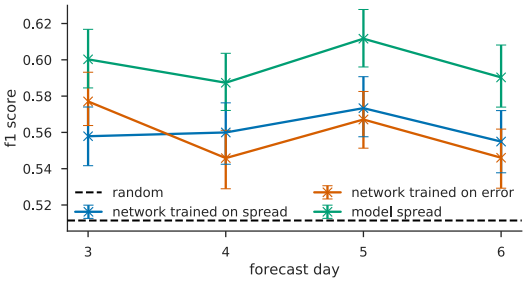


FIGURE 7 F1 score for classifying forecasts as having below or above average error anomalies for the network trained on spread (blue), the network trained on error (orange) and the forecast spread (green). The dashed line indicates the F1-score of random predictions. The error bars show the uncertainty (5-95) estimated via bootstrapping.

define an objective usefulness threshold, we compute the F1 score for random predictions. The random predictions are done via randomly sampling from the forecast error anomaly distribution, and have an F1-score of ∼0.51. This is congruent with the theoretical value of 0.5 for a perfectly balanced 2-class dataset. Thus all predictions of forecast error that have a score significantly higher than 0.51 are better than random. Better-than-random is a necessary but not sufficient condition for usefulness. It is impossible to give a single threshold for usefulness, as this is strongly dependent on the application. Better-than-random can, however, be used as a lower threshold, above which the method could in principle be useful for at least some applications. The results for the F1-score are shown in fig. 7. The dashed line indicates the skill of random predictions. While both CNN methods have less skill than the model spread (green line), both are significantly better than random for all lead-times. The scores split up according to season are shown in fig. 8 for forecast days 3 and 6. At day 3, the seasonal cycle in F1-score is less clear than the one for correlation with forecast spread. However, both methods have significant skill in all seasons at day 3. At forecast day 6 both methods are skillful for all seasons except JJA. Interestingly, the F1 score at day 6 peaks in JJA, contrary to the correlation, which has its minimum in JJA at day 6 (see fig. 4).

The F1-score provides an overview of the performance of the predictions in classifying forecasts as having below or above average uncertainty. However, it gives no information on how far off from the real error predictions are. Therefore, we also consider the skill of the CNNs in predicting that a certain forecast will have an error anomaly above a given threshold (e.g. above average error, or an error anomaly of above 1 standard deviation etc). For this we compute the false positive rate, the false negative rate, the true positive rate and the true negative rate for different thresholds. The result is shown in fig. 9. The splitting into false positive, false negative, true positive and true negative rates allows to deduce the implications in an operational context. For example, let us assume a user wishes to know whether a forecast belongs to the upper tercile in terms of uncertainty (forecast error anomaly > 1). The false negative rates in fig. 9 are ∼0.8 and the true negative rates are ∼0.88. This tells us that, while many forecasts are incorrectly classified as not exceeding the threshold, the predictions are still better than random.

To summarise: for all seasons and lead times, except for summer at lead times >3 days, both our methods are capable of predicting forecast uncertainty significantly better than random sampling. The spread of the weather forecast model is a better predictor of actual forecast skill than both our methods. However, after being trained (which has to be done only once), our method is computationally orders of magnitude cheaper, and even the training is signficiantly less expensive than running a NWP ensemble.

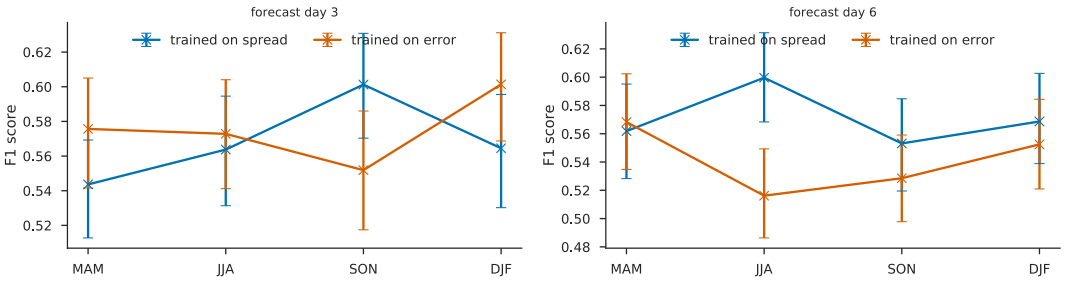


FIGURE 8 Seasonal F1 scores for classifying forecasts as having below or above average error anomalies, for the network trained on spread (blue) and trained on error (orange). The error bars show the uncertainty (5-95) estimated via bootstrapping.

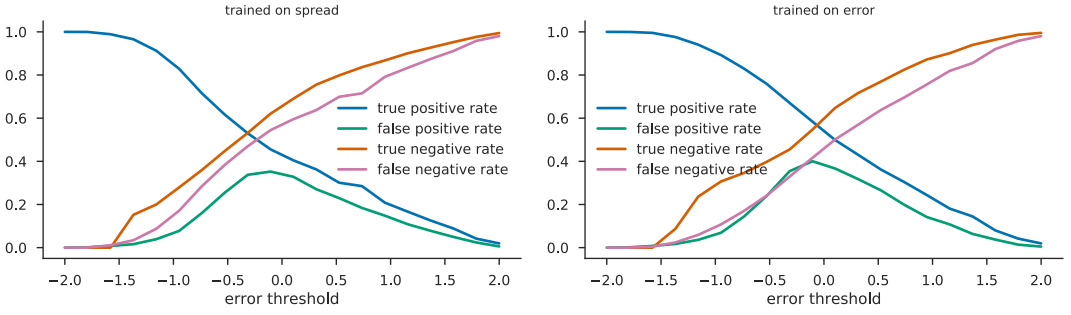


FIGURE 9 Skill in predicting forecast error above a given threshold, for the network trained on spread (left) and trained on error (right), for forecast day 3.

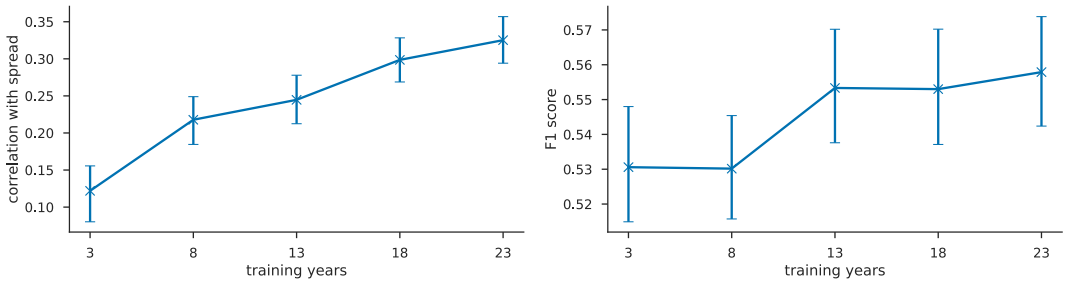


FIGURE 10 Correlation between predicted uncertainty and model spread (left) and F1-score for predicting higher/lower error than normal for different lengths of training data (right), for forecast day 3, trained on spread. The error bars show the uncertainty (5-95) estimated via bootstrapping.

4.3 | Dependence on length of training period

As our method is purely data-driven, it is of interest to know how sensitive it is to the amount of available training data. Therefore, we now test how dependent the skill of our method is on the length of the training dataset. For this, we gradually decrease the length of the training set, from 23 to 3 years, in steps of 5 years. We always use the final years of the training data. This corresponds to an operational setting, where reforecasts will usually be available from some point in the past up to near-present. The validation data was kept the same as in our standard training dataset. The analysis was done for forecast day 3 and trained on forecast spread. The skill of our method - computed both as the correlation between predicted uncertainty and ensemble spread and as F1 score for classifying forecast error - clearly decreases with decreasing training data (fig. 10). The correlation decreases from ∼0.35 with the original 23 years of training data to close to 0.1 with only 3 years. Similarly, the F1-score decreases from ∼ 0.56 for the original training data down to ∼0.53 with only 3 years. Moreover, while there is a leveling off of the CNN's performance on the higher end of the plots, no plateau seems to be reached yet. This suggests that, with longer training sets which might become available in the future, the skill of our method should increase further. In fact, in contrast to the sensitivity to the length of the training periods, testing slightly different input domains and variables lead to only small changes in the predictive skill (see Section 2). This suggests that the main limitation of our method is the amount of available past forecasts.

Note that we do not use the last 6 years of the available data for training, because we want to have an extensive testing data set. However, in an operational context, all available years could be used for training (excluding the 2 years necessary for validation and tuning). Thus, in such an operational setting, the skill would probably be slightly higher than what shown here.

4.4 | Comparison to baseline methods

In order to assess the usability of our method and place it in a broader context, we now compare our results to the baseline methods described in the Section 3.2. We compare the correlation of the predicted uncertainties from our networks and the baseline methods with the GEFS reforecast ensemble spread, for lead days 3-6. Additionally, we computed the F1 score for classifying forecast error. The results are shown in fig. 11. Our method significantly outperforms all the other methods at most lead-times, both in terms of correlation with forecast spread and in classifying forecast error. The nearest-neighbour approach shows a significant but very low correlation with the ensemble spread for days 3 and 4, and no significant correlation at days 5 and 6. The local dimension (d) shows significant but low correlation at all lead times except for day 6. Clustering does not work well for short lead times, but it approaches the skill of the neural network trained on spread at day 6. θ is clearly the best performing of the baseline methods, approaching the skill of the networks at day 4 and 6. In terms of the F1-score, our two

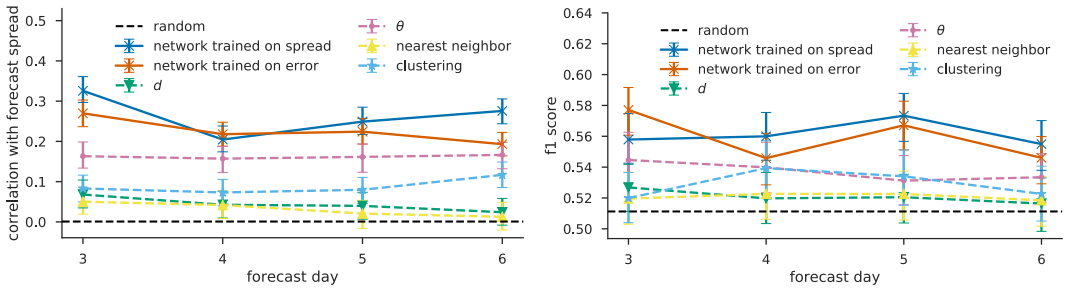


FIGURE 11 Comparison of our method with 4 baseline methods: nearest neighbor (yellow), weather type clustering (turquoise), local dimension (d, green) and the inverse of persistence in phase space (θ, pink). Shown is the correlation coefficient with the model spread (left) and F1-score for classifying forecast error as being above or below average (right). The error bars show the uncertainty (5-95) estimated via bootstrapping.

methods are the best for all days. Similar to correlation, clustering and θ are the best performing baseline methods.

We note that the correlation values of d and θ with forecast spread found here are lower than those reported by Faranda et al. (2017). In their analysis, the authors removed the seasonal cycle of d, θ and the forecast spread by subtracting a 180 day running mean, which retains some seasonality. Here, we have instead computed the anomalies via subtracting a 30 day running mean climatology (see Method section). Since the three variables share a similar seasonal cycle, this methodological difference explains the discrepancy in the reported values. Finally, we also note that the domains used for the nearest neighbour method, d and θ were optimized for the neural network methods, which might penalize the alternative methods.

5 | CONCLUSIONS

We have evaluated whether it is possible to use machine learning techniques in order to infer weather forecast uncertainty from the error and spread of past ensemble forecasts. We showed that this is possible with a method that is based on a convolutional neural network, which takes as input atmospheric fields and is trained with either the error of past deterministic weather forecasts, or the ensemble spread of past ensemble forecasts. When presented with a new atmospheric field (one that was not used during the training), it assigns a scalar value of predictability (or forecast uncertainty) to this field. Both our networks are shown to be useful predictors of forecast uncertainty. The input data used (GEFS reforecast v2) are being updated on a daily basis. Thus, our method could directly be applied in an operational setting. Our method is not as skillful in predicting forecast errors as the spread of ensemble forecasts. However, at almost all lead times up to 6 days it significantly outperforms two other methods that have been proposed in the literature, namely clustering by weather type by Ferranti et al. (2015), and persistence in phase space by Faranda et al. (2017). There is nonetheless the possibility, not verified here, that at longer lead times these alternative methods may perform better than machine learning. The main limitation of our method seems to be the amount of available past forecasts.

In terms of computational efficiency, training the network takes around 30 minutes on 2 NVidia K80 GPUs. Inference at testing time (presenting the network with an input field and predicting the uncertainty) is on the order of seconds, and thus orders of magnitude cheaper than running an ensemble NWP model. In an operational context, the network could be retrained every day (when the error of a new forecast becomes available), in order to always use the largest possible amount of training data. One could, for example, run the network prediction before running an operational ensemble forecast model, and use the predicted uncertainty to decide how many ensemble members are necessary. In less predictable situations it may make sense to invest more computation time and run more members, whereas in highly predictable situations fewer members may be sufficient,

and the saved computation time could be used for other tasks. Additionally, due to the comparative inexpensiveness of our method, it could be applied to climate studies. One could train it once on past forecasts, and then determine the predictability of the atmosphere for each day in a long climate model simulation. This would allow assessing whether there are low-frequency variations in predictability, or whether there are systematic changes in connection with climate change. Performing such an analysis with an ensemble forecast model would be computationally extremely expensive.

We have to note that ensemble forecasts contain more information than only an uncertainty measure. They can be used to compute the likelihood of certain events, or to create individual scenarios based on single members. Therefore, our proposed methods are not to be seen as a replacement of ensemble forecasts, but rather as tools that exploit already existing past forecasts to provide complementary guidance. To conclude, we note that this study is to be seen as a proof-of-concept to show that it is possible to extract information on forecast uncertainty from past forecasts with machine learning techniques. Indeed, there are a number of potential ways to improve the skill of the methods proposed here. One could for example extend reforecast datasets to earlier dates to overcome the limitations of available past forecasts, combine different reforecast dataset for training or try to blend in old operational forecasts. It would also be very interesting to extend the methods to different scales both in space (e.g. regional weather models) and time (sub-seasonal and seasonal prediction). From the usage point of view, it is also interesting to assess the possibility of training explicitly on predicting very high forecast error events.

DATA AVAILABILITY

The data of the GEFS reforecasts v2 is available from NOAA at https://www.esrl.noaa.gov/psd/forecasts/reforecast2/download.html

The output data from our methods can be obtained from S.S. on request.

The used software libraries (Keras and Tensorflow) are open source and available on https://pypi.python.org/pypi/

AUTHOR CONTRIBUTION STATEMENT

S. Scher has designed the study, developed and implemented the machine learning methods, analyzed the results and prepared the manuscript. G. Messori prepared the d and θ data and contributed to analyzing the results and drafting the manuscript.

ACKNOWLEDGEMENTS

The computations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at the High Performance Computing Center North (HPC2N) and National Supercomputer Centre (NSC).

We want to thank Ling Yiu for his helpful tips on using CNNs and machine learning. Further, we thank Erland Källén for helpful discussions during the initial phase of this study, and Rodrigo Caballero for his input when analyzing the results. S. Scher and G. Messori have been funded by the Department of Meteorology of Stockholm University and by Vetenskapsrådet grant no. 2016-03724.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

ENDNOTES

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. and Zheng, X. (2015) TensorFlow: Large-scale machine learning on heterogeneous systems. URL: `https://www.tensorflow.org/`. Software available from tensorflow.org.

Anagnostou, E. N. (2004) A convective/stratiform precipitation classification algorithm for volume scanning weather radar observations. *Meteorological Applications*, **11**, 291–300.

Bankert, R. L. (1994) Cloud Classification of AVHRR Imagery in Maritime Regions Using a Probabilistic Neural Network. *Journal of Applied Meteorology*, **33**, 909–918. URL: `https://journals.ametsoc.org/doi/abs/10.1175/1520-0450(1994)033%3C0909:CCOAII%3E2.0.CO%3B2`.

Bauer, P., Thorpe, A. and Brunet, G. (2015) The quiet revolution of numerical weather prediction. *Nature*, **525**, 47–55. URL: `https://www.nature.com/articles/nature14956`.

Chollet, F. et al. (2015) Keras. `https://github.com/keras-team/keras`.

Dibike, Y. B. and Coulibaly, P. (2006) Temporal neural networks for downscaling climate variability and extremes. *Neural Networks*, **19**, 135–144. URL: `http://www.sciencedirect.com/science/article/pii/S0893608006000062`.

ECMWF (2018) ECMWF Charts. URL: `https://www.ecmwf.int/en/forecasts/charts/catalogue/`.

Faranda, D., Messori, G. and Yiou, P. (2017) Dynamical proxies of North Atlantic predictability and extremes. *Scientific Reports*, **7**, 41278. URL: `https://www.nature.com/articles/srep41278`.

Ferranti, L., Corti, S. and Janousek, M. (2015) Flow-dependent verification of the ECMWF ensemble over the Euro-Atlantic sector. *Quarterly Journal of the Royal Meteorological Society*, **141**, 916–924. URL: `http://onlinelibrary.wiley.com/doi/10.1002/qj.2411/abstract`.

Ghosh, S. and Mujumdar, P. P. (2008) Statistical downscaling of GCM simulations to streamflow using relevance vector machine. *Advances in Water Resources*, **31**, 132–146. URL: `http://www.sciencedirect.com/science/article/pii/S0309170807001224`.

Gneiting, T. and Raftery, A. E. (2005) Weather forecasting with ensemble methods. *Science*, **310**, 248–249.

Hamill, T. M., Bates, G. T., Whitaker, J. S., Murray, D. R., Fiorino, M., Galarneau, T. J., Zhu, Y. and Lapenta, W. (2013) NOAA's Second-Generation Global Medium-Range Ensemble Reforecast Dataset. *Bulletin of the American Meteorological Society*, **94**, 1553–1565.

Hewitson, B. C. and Crane, R. G. (1992) Large-scale atmospheric controls on local precipitation in tropical Mexico. *Geophysical research letters*, **19**, 1835–1838.

Hsieh, W. W. and Tang, B. (1998) Applying Neural Network Models to Prediction and Data Analysis in Meteorology and Oceanography. *Bulletin of the American Meteorological Society*, **79**, 1855–1870.

Kingma, D. P. and Ba, J. (2014) Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*. URL: `http://arxiv.org/abs/1412.6980`. ArXiv: 1412.6980.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012) ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25* (eds. F. Pereira, C. J. C. Burges, L. Bottou and K. Q. Weinberger), 1097–1105. Curran Associates, Inc. URL: `http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf`.

Lakshmanan, V., Karstens, C., Krause, J. and Tang, L. (2014) Quality Control of Weather Radar Data Using Polarimetric Variables. *Journal of Atmospheric and Oceanic Technology*, **31**, 1234–1249. URL: https://journals.ametsoc.org/doi/abs/10.1175/JTECH-D-13-00073.1.

LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep learning. *Nature*, **521**, 436–444. URL: https://www.nature.com/articles/nature14539.

Liu, Y., Racah, E., Prabhat, Correa, J., Khosrowshahi, A., Lavers, D., Kunkel, K., Wehner, M. and Collins, W. (2016) Application of Deep Convolutional Neural Networks for Detecting Extreme Weather in Climate Datasets. *arXiv:1605.01156 [cs]*. URL: http://arxiv.org/abs/1605.01156. ArXiv: 1605.01156.

Lorenz, E. N. (1963) Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences*, **20**, 130–141.

Marzban, C. and Stumpf, G. J. (1996) A Neural Network for Tornado Prediction Based on Doppler Radar-Derived Attributes. *Journal of Applied Meteorology*, **35**, 617–626. URL: https://journals.ametsoc.org/doi/abs/10.1175/1520-0450(1996)035%3C0617:ANNFTP%3E2.0.CO%3B2.

McGovern, A., Elmore, K. L., Gagne, D. J., Haupt, S. E., Karstens, C. D., Lagerquist, R., Smith, T. and Williams, J. K. (2017) Using Artificial Intelligence to Improve Real-Time Decision-Making for High-Impact Weather. *Bulletin of the American Meteorological Society*, **98**, 2073–2090. URL: https://journals.ametsoc.org/doi/abs/10.1175/BAMS-D-16-0123.1.

Pasini, A., Racca, P., Amendola, S., Cartocci, G. and Cassardo, C. (2017) Attribution of recent temperature behaviour reassessed by a neural-network method. *Scientific Reports*, **7**, 17681. URL: https://www.nature.com/articles/s41598-017-18011-8.

Rodwell, M. J., Magnusson, L., Bauer, P., Bechtold, P., Bonavita, M., Cardinali, C., Diamantakis, M., Earnshaw, P., Garcia-Mendez, A., Isaksen, L., Källén, E., Klocke, D., Lopez, P., McNally, T., Persson, A., Prates, F. and Wedi, N. (2013) Characteristics of Occasional Poor Medium-Range Weather Forecasts for Europe. *Bulletin of the American Meteorological Society*, **94**, 1393–1405. URL: http://journals.ametsoc.org/doi/abs/10.1175/BAMS-D-12-00099.1.

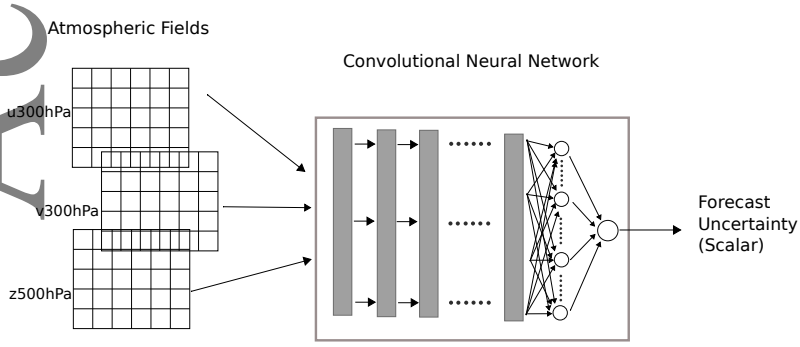
Schmidhuber, J. (2015) Deep learning in neural networks: An overview. *Neural Networks*, **61**, 85–117. URL: http://www.sciencedirect.com/science/article/pii/S0893608014002135.

Simmons, A. J. and Hoskins, B. J. (1979) The Downstream and Upstream Development of Unstable Baroclinic Waves. *Journal of the Atmospheric Sciences*, **36**, 1239–1254. URL: https://journals.ametsoc.org/doi/abs/10.1175/1520-0469(1979)036%3C1239:TDAUDO%3E2.0.CO%3B2.

Tangang, F. T., Tang, B., Monahan, A. H. and Hsieh, W. W. (1998) Forecasting ENSO Events: A Neural Network–Extended EOF Approach. *Journal of Climate*, **11**, 29–41. URL: https://journals.ametsoc.org/doi/full/10.1175/1520-0442(1998)011%3C0029:FEEANN%3E2.0.CO;2.

UNISDR (2007) Hyogo Framework for Action 2005-2015: Building the resilience of nations and communities to disasters - Extract from the final report of the World Conference on Disaster Reduction. *Tech. rep.*, United Nations Office for Disaster Risk Reduction (UNISDR).

GRAPHICAL ABSTRACT



Sketch of the implemented machine-learning method. Left the input data (2D atmospheric fields), in the center the convolutional neural network, and to the right the output (one scalar value representing atmospheric predictability).