

UNIVERSITY OF DEBRECEN
Faculty of Informatics
Department of Applied Mathematics and Probability Theory

Postprocessing visibility ensemble forecasts using Neural Networks

Tudományos Diákköri Dolgozat

Consultant:
Dr. Sándor Baran
Full professor

Author:
Gréta Tisza
Computer Science MSc II.

Debrecen, 2023

Bírálatra való beküldéséhez hozzájárulok:

Contents

1	Introduction	3
2	EPS and post-processing methods	6
2.1	Motivation	6
2.2	Ensemble Prediction System	8
2.3	Ensemble Postprocessing	9
2.3.1	Statistical postprocessing	10
2.3.2	Neural Network based approaches	12
2.4	Verification scores	14
3	Data / Predicting visibility	18
3.1	Motivation	18
3.2	The ECMWF EPS	18
3.3	Data	19
3.4	Data processing	20
4	The models	22
4.1	The linear model	22
4.1.1	Data	22
4.1.2	Architecture	23

<i>CONTENTS</i>	2
4.1.3 The training process	23
4.1.4 Hyperparameter fine-tuning	24
4.2 The convolutional model	25
4.2.1 Data	25
4.2.2 Architecture	25
4.2.3 The training process	26
4.2.4 Hyperparameter fine-tuning	27
5 Results	30
6 Discussion	36
7 Conclusion	37
8 Own work	39

Chapter 1

Introduction

Visibility (the greatest distance at which an object can be seen and recognized) plays a crucial role in aviation in all phases of flight. Not having good visibility conditions causes serious safety risks in aviation, but restricted visibility also makes other types of transportation difficult. Hence, the reliability of visibility forecasts have both direct economic effect and a significant role in safety.

The development of accurate weather prediction systems is challenging due to the complicated nature of the atmosphere and the chaotic behavior of its elements. Minor inaccuracies in observations and forecast model configurations propagate chaotically, resulting in significant prediction errors over time. Predicting extreme, rare events is also a complicated task. Numerical Weather Prediction (NWP) systems are based on computer models solving complex partial differential equations. The complexity of the simulation and the spatial resolution are both limited by computational power. Small errors can be introduced through spatial and temporal discretization, and through empirical formulations for unresolved processes (which are too complex for simulation). These errors are amplified due to the differential equations' sensitivity to initial conditions, therefore, forecasts are loaded with uncertainty. To quantify forecast uncertainty, Ensemble Prediction Systems (EPSs) generate probabilistic predictions by running ensembles of deterministic NWM models that differ in initial conditions and/or model parametrization. (Buizza, 2018)

Most NWP models do not model visibility explicitly, in contrast to other quan-

tities like temperature or precipitation. These systems derive visibility predictions from forecasts of related weather quantities, such as relative humidity, cloud water content or precipitation. (Chmielecki and Raftery, 2011) One exception is the European Centre for Medium-Range Forecasts (ECMWF) which provides visibility ensemble forecast since 2015. (ECMWF, 2021)

To achieve proper calibration (consistency between the forecasts and the observations) ensemble forecasts often need postprocessing for bias correction and dispersion adjustments, especially for visibility forecasts, which tend to perform highly below the skill of ensemble forecast of e.g temperature or wind speed.

In the last decades various postprocessing methods were developed for different weather quantities. There are parametric approaches providing a full predictive distribution of the investigated weather quantity like non-homogeneous regression or Bayesian model averaging, or nonparametric methods estimating the quantiles of the forecast distribution. An overview about state-of-the-art postprocessing methods can be found in Vannitsem *et al.* (2018). In the recent years machine learning (ML) also gains popularity in the field. A detailed comparison of neural network based methods can be found at Schultz and Lerch (2022). Recently deep learning was also used in some experiments with promising results, see Grönquist *et al.* (2021).

Despite the continuous nature of visibility, most of synoptic surface observations are reported according to the World Meteorological Organization (WMO) standard 84 categories. Hence, predicting visibility categories can be considered as a classification problem. In this work we even more simplify the observations and consider only 10 categories.

In this work we explore two different models for directly predicting the probability distribution over the categories from the ensemble data. The first model is a linear MLP model operating only from derived features. The second one is a convolutional neural network (CNN) which extracts additional features from the raw ensembles and from the observational data of previous days before classification. The aim of this work is to explore if convolutional layers can recognize patterns in the raw ensembles or in the historical data to make future predictions more accurate.

Chapter 2 introduces ensemble forecasting and the motivation behind it. A few statistic-based and neural network based postprocessing methods are also presented in this chapter, together with the most popular scoring rules for forecast evaluation. A brief description about the data used in this work can be found in Chapter 3. Chapter 4 presents the model and describe its training process. The results are presented in Chapter 5 and discussed in Chapter 6. Chapter 7 summarizes this work. My own work in the experiment is described in Chapter 8.

Chapter 2

EPS and post-processing methods

2.1 Motivation

Numerical weather prediction (NWP) is a computer-based forecast method. It predicts a future state of the atmosphere by using mathematical equations that describe the physical processes happening in the atmosphere, based on a combination of atmospheric observations and information about the underlying surface.

NWP models are subject to various sources of uncertainty.

- **Limitations of the model itself**

Representing the atmosphere and describing the processes within it involves many simplifications by replacing physical processes that are too small-scale or complex to be explicitly resolved in the model. Such a technique is using parametrization schemes to approximate the effects of the small-scale processes on the larger-scale flow of the atmosphere, where uncertainty comes from unresolved physical processes (e.g. cloud microphysics).

- **Amplified errors in the initial conditions**

The accuracy of the forecast depends on the quality and the quantity of data that is used to initialize the model. In the estimation of the starting conditions uncertainty arises due to the limited number of observed variables and the observational error. Besides the measurement errors caused by

the possible imperfections in the calibration of the measuring devices, some variables are inferred indirectly (e.g. satellite temperature measurements), which leads to more risk of errors.

Weather is inherently chaotic, meaning that small changes in the initial conditions can lead to large differences in the forecast. The small errors in the initial conditions are amplified by the chaotic nature of the evolution equations predicting the behaviour of the atmosphere. This behaviour is often referred to as sensitive dependence on initial conditions (butterfly effect), which is a main characteristic of nonlinear dynamic systems undergoing deterministic chaos. (Lorenz, 1963, 1993)

- **Model Resolution**

NWP models divide the atmosphere into a 3D system of grid cells, whose size determines the resolution of the model. Higher resolution can capture more details and result more accurate forecasts, but it also requires more computational power and resources. Lower resolution can result lower accuracy, especially for small-scale weather events. Since it is not possible to collect infinitely detailed information, uncertainty arises.

- **Boundary Conditions**

The behavior of the atmosphere is also affected by the regions outside of the covered volume. The conditions at its boundaries can be based on observations, or they might be derived from larger-scale models. Uncertainty can introduced by inaccuracy or incompleteness in the data (caused by the difference of spatial resolution or measured variables between the model and the data coming from outside sources), together with errors propagated from the outer models.

However, the uncertainty can be approximated by probability distributions. Several techniques to represent a model's uncertainty have been proposed which - in general - introduce random variability into the model. Widely used examples are the perturbed parameter technique (randomly varying the model's parameters to explore the range of possible states), stochastic parametrization (using stochastic differential equations to catch how variables evolve in response to random perturbations (Christensen *et al.*, 2015)), and ensemble forecasting (which varies the

initial conditions and/or model parametrization to assess the likelihood of different outcomes).

2.2 Ensemble Prediction System

Ensemble Prediction Systems (EPS) are numerical weather prediction (NWP) systems that let us assessing the uncertainty coming from the (above mentioned) amplified errors in the initial conditions. Instead of running the NWP model once (a deterministic forecast) with the single observed initial state, the model is run many times with slightly different initial conditions. Each modified state differs from the original observation only in the extent to reflect possible errors or inadequacies in observation (Buizza, 2018). By applying the model to the perturbed initial conditions an ensemble of future states is generated, from which the probability of occurrence of an event can be approximated better than by deriving it from only one state.

Therefore the EPS method is used to provide probabilistic forecasts. Most operational EPSs provide a control forecast that uses the model without any perturbations to the initial conditions, and the ensemble members, the individual NWP solutions resulted from the slightly changed initial conditions. The set of different predicted future states give opportunity to estimate uncertainty in the forecast and let us infer our confidence in the original, deterministic forecast (WMO, 2012). Small ensemble dispersion should indicate smaller uncertainty, meaning that if all members give similar results, we can be highly confident in the forecast. In the same way, large ensemble dispersion corresponds to larger forecast uncertainty, meaning that substantial differences among ensemble members indicate less confidence in the forecast, and the ensemble can be used to approximate the probabilities of various future states (WMO, 2021).

Ensemble members obtained using perturbed initial conditions are statistically indistinguishable, thus can be considered as exchangeable. On contrary, there are also EPSs where one can clearly track each ensemble member separately. Ensemble prediction is a type of Monte-Carlo methods which is described in details in the influential paper by Leith (1974).

Ensemble forecasting was first implemented in a meteorological context by Epstein (1969). Ensemble forecasting was first used operationally in 1992 at the European Centre for Medium-Range Weather Forecasts and at the U.S. National Meteorological Center (Molteni *et al.*, 1996).

2.3 Ensemble Postprocessing

Since the forecast ensemble is a random sample from the forecast-uncertainty distribution, the probability of an event can be estimated from the proportion of ensemble members predicting that very event (Vannitsem *et al.*, 2018).

For making accurate predictions EPSs should be able to sample all potential sources of forecast error. Simply using random perturbations of the initial conditions is not sufficient to sample and represent all possible errors. For instance, forecasts can be biased due to inaccuracies and incompleteness in the formulation of the simulations (Vannitsem *et al.*, 2018). Therefore, postprocessing is required to correct systematic bias and for making adjustments to the dispersion in order to achieve consistency between the forecasts and the observations.

As mentioned, over the past decade, various statistical postprocessing methods have been proposed. These methods vary in complexity, required computing resources, the type and amount of data, and also in technical skills necessary to setup and operate the system. Based on these requirements they are categorized into 3 tiers by the WMO. Conceptually intuitive systems which can be implemented even on a desktop are classified as Tier 1. They are typically single-site, univariate approaches, and work with relatively short historical records (less than a few years). In Tier 2 there are methods which possibly involve multivariate inputs and require some scientific testing skills for performance optimization. They usually work with larger datasets and gridded data. (Some Tier 1 algorithms might also require Tier 2 computational capabilities on high-resolution gridded data.) Tier 3 techniques involve advanced mathematical methods and a detailed meteorological understanding might also be required (WMO, 2021).

Modern EPSs require vast computing resources for operation, moreover, developing and improving them is also extremely expensive. Postprocessing systems

are designed to enhance already existing ensemble forecasts and they can result in substantial improvements, especially for regional forecasts, while they are more economical to develop and use.

2.3.1 Statistical postprocessing

The objective of statistical postprocessing methods is to analyze and statistically describe errors present in a dataset of previous predictions generated by a specific system. This information is then leveraged to correct future forecasts produced by the same system.

Nonhomogeneous regression

One of the two most frequently used statistical postprocessing methods is Nonhomogeneous regression, which is a regression-based method that fits a probability distribution to the ensemble forecast. The type of the used distribution depends on the behaviour of the examined weather quantity. The mean of the predictive distribution is usually defined as an optimized linear combination of the ensemble members and the variance depends linearly on the ensemble variance. As a result of this property, the predictive distributions are capable of demonstrating varying levels of uncertainty - when the ensemble dispersion is large, they can exhibit larger uncertainty, whereas with the decrease of the ensemble dispersion the uncertainty also decreases (Vannitsem *et al.*, 2018).

One of the most commonly chosen distribution is the Gaussian distribution. Nonhomogeneous Gaussian regression (NGR) was proposed independently by Gneiting *et al.* (2005) and Jewson *et al.* (2004) and was named Ensemble Model Output Statistics (EMOS); however, it is not the only EMOS method existing.

Some meteorological variables are not Gaussian or approximately Gaussian, so that the NGR has to be modified to describe them adequately. Several approaches were proposed, e.g.

- applying Box-Cox transformation on the ensembles before fitting (Box and Cox, 1964),

- using truncated Gaussian predictive distributions for forecasting positively skewed and nonnegative variables, such as wind speed (Thorarinsdottir and Gneiting, 2010),
- using a censored distribution, which allows modelling a point mass representing e.g. the probability of zero precipitation (Scheuerer, 2014; Scheuerer and Hamill, 2014; Baran and Nemoda, 2016) or zero solar irradiance (Schultz *et al.*, 2021)
- or combining predictive distributions, like truncated normal and log-normal distributions, as in the work of Baran and Lerch (2016).

Bayesian model averaging

The second of the two most commonly used statistical postprocessing methods is Bayesian model averaging (BMA) proposed by Raftery *et al.* (2005). Rather than imposing a particular parametric form, BMA predictive distributions are mixtures of distributions each centered at the bias corrected value of one of the m ensemble members being postprocessed. A BMA predictive distribution can be expressed as:

$$p(x) = \sum_{k=1}^m w_k g(x|f_k, \theta_k)$$

where each w_k is the nonnegative weight associated with the component probability density $g(x|f_k, \theta_k)$ corresponding to the k th ensemble member f_k with parameter vector θ_k . The weights should be from a probability distribution, therefore $\sum_{k=1}^m w_k = 1$.

This approach has been used widely for various weather quantities, see e.g. Sloughter (2010) or Baran (2014).

Proportional odds logistic regression

If the predictable variable is categorized and the categories are ordered (such as in the case of discrete weather quantities), proportional odds logistic regression (POLR) also can be used. Consider a discrete random variable Y and an M -dimensional feature vector $\mathbf{x}$ derived from the raw ensemble. The conditional

cumulative distribution of Y with respect to $\mathbf{x}$ is given as:

$$P(Y \leq y_k | \mathbf{x}) = \frac{e^{\mathcal{L}_k(\mathbf{x})}}{1 + e^{\mathcal{L}_k(\mathbf{x})}}$$

with $\mathcal{L}_k(\mathbf{x}) = \alpha_k + \mathbf{x}^\top \boldsymbol{\beta}$, where y_k is the observation corresponding to k th category, and k goes from 1 to the number of categories. The parameters α_k follow an ascending order and together with $\boldsymbol{\beta}$ are estimated from the training data consisting of past forecast-observation pairs (Baran *et al.*, 2021).

Quantile regression

Quantile regression is another ensemble postprocessing method introduced by Koenker *et al.* (1978). In this approach, the relation between the observations and the ensemble is modeled only for a limited number of quantiles, usually individually for each considered quantile. Instead of estimating the conditional mean of the predictive distribution, quantile regression estimates the conditional quantile functions through regressing the conditional median. It was first applied to weather forecasting by Bremnes (2004).

2.3.2 Neural Network based approaches

Recently, machine learning (ML) based techniques gain more and more popularity probabilistic weather forecasting. Neural Network (NN) based methods can explore new relations in the data and might give more reliable results. This field is still under research, but the results are promising. Some examples from the last decades are:

Distributional Regression Network

Rasp and Lerch (2018) propose a postprocessing method based on NNs that extends the EMOS framework. They used a simple linear model and a model with one hidden layer for predicting the mean and variance of the predictive Gaussian distribution used in NGR EMOS on a 50-member ensemble. Instead of the raw ensemble they used a vector of predictors that only contains the mean value and

standard deviation of the forecasts. To enable the networks to learn station-specific information, station-embeddings were also concatenated to the base predictors, which resulted an improvement of 29% compared to the raw ECMWF ensemble forecasts in terms of the Continuous Ranked Probability Score (CRPS, see Chapter 2.4). Note that without station embeddings the network behaved as the basic NGR method, and gave the same improvement of 13%. The model with a single hidden layer did not show any improvements over the simple linear model.

Bernstein Quantile Network

Bremnes (2020) used neural networks in the quantile regression framework on a 50-member ensemble (+ control). A linear combination of Bernstein polynomials was employed to model the quantile function of the forecast distribution and their coefficients were estimated by a NN. Neural networks with one and two hidden linear layers were considered with number of units between 8 and 64, while the degree of Bernstein polynomials was 4, 6 or 8 with 52 quantiles. The improvement between the raw ensembles and the output of the quantile regression method were significant; however, NN predicted quantiles showing decrease in the considered skill score compared to the statistically inferred quantiles.

Convolutional networks

The above mentioned methods used neural networks (containing only linear layers) in the regression phase of statistics-based methods for calculating coefficients and parameters of the statistical model. However, deep learning can exploit connections in the data in new ways. Convolutional Neural Networks (CNN) are especially made for pattern recognition, primarily for extracting spatial information, but they used in other various fields such as natural language processing or processing time series. CNN-s were used in some experiments with weather forecasting in the recent years (see e.g. Weyn *et al.*, 2020), but in very few cases concerning ensemble postprocessing.

One example is the work of Grönquist *et al.* (2021), in which two different models were built for uncertainty quantification and output bias correction tasks. They

used Inception-style modules (Szegedy *et al.*, 2015) and a U-Net (Ronneberger *et al.*, 2015) architecture with residual connections passing unmodified features between layers that are not directly connected to each other. On average, their model improved 14.5 % to the raw ensemble and 9% compared to the EMOS NGR method. For extreme events the model brought larger improvement, more than 25.5% compared with the raw ensemble.

2.4 Verification scores

Probabilistic forecasts of a continuous variable usually take the form of predictive densities or predictive cumulative distribution functions. In probabilistic forecasting the main objective is to reach the maximal sharpness of the predictive distribution while ensuring calibration. Calibration is a common property of the predictions and the observations and it means the statistical consistency between them. Sharpness refers to the concentration of the predictive distributions, which is independent from the observations (Gneiting *et al.*, 2007).

To assess calibration and sharpness at the same time, one can utilize proper scoring rules. They provide summary metrics in the form of a numerical score assigned to a forecast based on the predictive distribution and the corresponding observed event.

Denote by $S(P, x)$ the score for a predictive distribution P and an observation x . The function $S(P, \bullet)$ takes values in the real line $\mathbb{R}$ or in the extended real line $\overline{\mathbb{R}} = [-\infty, \infty]$, and we write $S(P, Q)$ for the expected value of $S(P, \bullet)$ under x following distribution Q . A scoring rule is said to be proper if $S(Q, Q) \geq S(P, Q)$ for all P and Q . It is said to be strictly proper if $S(Q, Q) = S(P, Q)$ if and only if $P = Q$ (Gneiting and Raftery, 2007).

Continuous Ranked Probability Score

One of the most popular scoring rules for density forecasts is the continuous ranked probability score (CRPS) (Wilks, 2019), which is defined directly in terms of the

predictive cumulative distribution function F as:

$$CRPS(F, y) = \int_{-\infty}^{\infty} [F(t) - H(t - y)]^2 dt$$

where $H(t - y)$ is the Heaviside step function which takes the value 0 when $t < y$ and the value 1 otherwise.

An alternative representation is:

$$CRPS(F, y) = E|X - y| - \frac{1}{2}E|X - X'| \quad (2.1)$$

where X and X' are independent copies of a random variable with cumulative distribution function F and finite first moment (Gneiting and Raftery, 2007).

The CRPS is proper and it generalizes the absolute error, to which it reduces if F represents a point forecast. Further, CRPS is a negatively oriented score, smaller values indicate better predictive performance. It behaves as a loss function assigning numerical values to forecast-observation pairs (F, x) , and we can rank competing forecasts based on the average CRPS over the verification data (Gneiting *et al.*, 2007).

Predicting visibility categories is a classification problem, so the predictive distribution is not a continuous law but probabilities of discrete classes. Therefore, we use the discrete version of the representation (2.1).

$$CRPS(F, x) = \sum_{k=1}^n p_F(y_k) |y_k - x| - \sum_{k=2}^n \sum_{\ell=1}^{k-1} p_F(y_k) p_F(y_\ell) |y_k - y_\ell|$$

where p_F is the probability mass function corresponding to forecast the discrete predictive distribution F , y_k is the reported visibility representing category k , and n is the number of categories (Baran *et al.*, 2021).

Logarithmic Score

The logarithmic score or ignorance score is simply the negative of the logarithm of the predictive density evaluated at the observation. (Good, 1952)

For a discrete predictive distribution F and an observation x , the logarithmic score $LogS(F, x)$ is defined as:

$$LogS(F, x) = -\log(p_F(x)),$$

where p_F is the probability mass function corresponding to forecast F .

Another way of looking at this equation is as the negative of the log-likelihood function, thus, optimum score estimation based on the logarithmic score is simply a maximum likelihood estimation.

The logarithmic score is also negatively oriented, meaning that it ranges from 0 for perfect forecasts to ∞ for certainty forecasts that are wrong. Forecast probabilities evaluated using the logarithmic score are to be estimated on the basis of a finite sample (e.g., in the context of ensemble forecasting) (Wilks, 2019).

Note that if $p_F(y_k)$ is 0, the score is undefined. 0 probabilities should be avoided unless the event is truly impossible, thus they are replaced with small probabilities based on the uncertainties in the forecasting probability distribution. Not to do so would mean reporting the improbable as the impossible (Roulston and Smith, 2002).

Skill scores

Small differences in predictive performance between forecasts can be further investigated by looking at the improvements with respect to a reference forecasts (Gneiting and Raftery, 2007). Scores (e.g. CRPS or LogS) are usually aggregated over the verification data, so that forecasts can be compared by the average score. Consider the score S , and let $\overline{S_F}$ denote the mean score value for forecast F . The skill score SS derived from score S can be defined as

$$SS = 1 - \frac{\overline{S_F}}{\overline{S_{ref}}}$$

where S_F is the score corresponding to forecast F while S_{ref} corresponds to the reference forecast. Note that skill scores are positively oriented, the larger the better.

The PIT Histogram

The Probability Transform Integral (PIT) (Wilks, 2019) histogram is a graphical tool to analyse consistency between forecasts and observations. For an observation

x and cumulative probability distribution function F

$$PIT(x) = F(x),$$

where in the case of discrete distribution a randomization is required. If the forecast is calibrated, then the PIT values should follow a standard uniform distribution.

Chapter 3

Data / Predicting visibility for aviation purposes

3.1 Motivation

Meteorological visibility is defined as "the greatest distance at which a black object of suitable dimensions (located on the ground) can be seen and recognized when observed against the horizon sky during daylight" (WMO, 1966, 2010).

Visibility plays a critical role in aviation. Pilots need good visibility conditions for several reasons e.g. to see landmarks and other visual aids to navigate correctly, to see traffic signs or potential hazards during takeoff and landing. Not having good visibility conditions causes safety risks. Predicting visibility conditions allows airlines to operate more efficiently by reducing delays and avoid cancellations, therefore accurate and reliable visibility forecasts result in economic benefit.

3.2 The ECMWF EPS

The European Centre for Medium-Range Weather Forecasts (ECMWF) started to produce ensemble predictions in December 1992. (Molteni *et al.*, 1996) Ensemble-mean forecasts have met the expectations based on previous studies and have shown superior performance compared to traditional high-resolution deterministic

predictions.

Due to the high computational cost associated with running an NWP model multiple times, the EPS is usually run at around half the horizontal resolution of the equivalent deterministic model. The ECMWF global medium-range forecast is made up of a high-resolution forecast (HRES) and an ensemble of lower-resolution forecasts (ENS). The horizontal resolution of the ensemble members are 18 km with 62 vertical levels. Though, the ensemble system is used for extended-range (monthly) forecasts as well. Twice a week, the ENS is extended up to 32 days using an ensemble forecasts with half of the original horizontal resolution. (ECMWF, 2012)

The number of ensemble members is also limited by computational power. Today the ECMWF has the largest ensemble forecast that uses 50 ENS members generated using random perturbations in the initial conditions and one control forecast (CTRL) resulted from the original initial conditions.

3.3 Data

In this work we consider 51-member visibility ensemble forecast provided by the operational EPS of the ECMWF. Prediction horizons range from 6 hours to 240 hours having 6 hour steps between lead times. The observations were made at: 00:00, 06:00, 12:00 and 18:00 UTC. The first day of the dataset is 01-01-2020, the last one is 31-12-2021.

The observations are made by 98 SYNOP (surface synoptic observations) weather stations. Some stations follow the WMO category suggestions for visibility: "100 to 5 000 m in steps of 100 m, 6 to 30 km in steps of 1 km, and 35 to 70 km in steps of 5 km." (This scale allows visibility to be reported with a better resolution than the accuracy of the measurement since the errors of visibility measurements increase in proportion to the visibility.) (WMO, 2014)

In the dataset:

- 35 stations follow the WMO suggestion.
- 41 stations follow the WMO suggestions but introduce some new categories

outside of the WMO categories' range: 30, 50, 90 m from the lower end, or 75000 and 81900 m from the upper end.

- 9 stations introduce some arbitrary categories in the middle region, or contain some not frequent values with minimal difference from a WMO category which might be measurement or communication errors.
- 13 stations use a 10 m step-size in their whole range.

The dataset is almost complete as there are no missing observations at all but predictions are missing in two cases: the ensemble members for 21-12-2021 with lead time 114 hours and for 07-06-2021 with lead time 132 hours are missing.

3.4 Data processing

Though visibility is a continuous variable, the usage of categories leads us to a classification problem. Instead of using the 84 WMO suggested categories, we classify the observations to 10 categories. The categories were chosen based on three aspects:

- Aligning with the WMO directives

As mentioned above, the errors of visibility measurements increase in proportion to the visibility (WMO, 2014). Hence, WMO categories are closer to each other in the lower half of the covered range. Following that guideline the steps between the chosen 10 categories are also proportional to the visibility.

- Frequency of categories

After recoding all observations to the WMO categories the frequency of each category can be measured. As figure 1 shows, a large proportion of the observations fall into the lower half of the covered range. Even though WMO reports until 70 km, the last frequent category is 50 km.

- Critical values

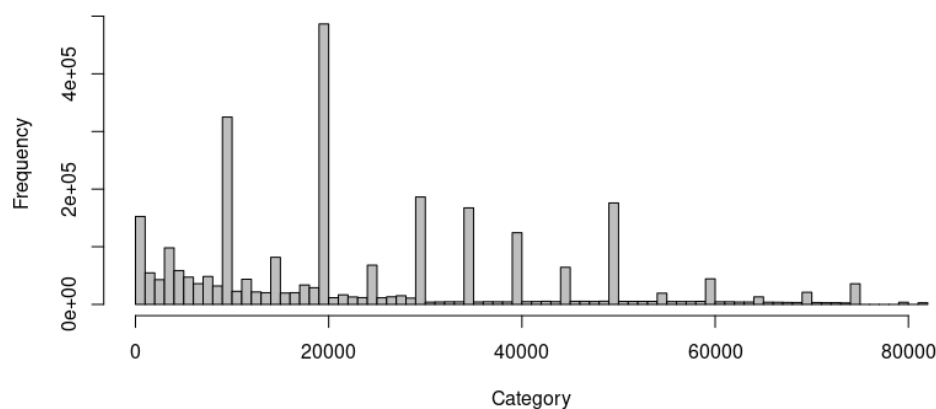


Figure 1: Frequency of WMO categories in the observations

Some stations report less than 10 different values, which mostly cover the 0-20 km range. For preserving the most information after recoding the intersection of these values are kept as categories, namely: $\{200, 1000, 4000, 10000\}$.

Considering the aspects mentioned above the following categories were chosen: $\{0, 50, 200, 400, 1000, 2000, 4000, 10000, 20000, 50000\}$.

All observations are recoded to these 10 categories. Every value is rounded down to the closest category.

Chapter 4

The models

In this work we try to take a step towards an end-to-end NN-based approach and build models to directly generate a probability distribution over the discrete categories $\{0, 50, 200, 400, 1000, 2000, 4000, 10000, 20000, 50000\}$. For each lead time a separate classification model is trained.

Two different types of architectures were tested. The first one is a simple linear MLP which operates on features derived from the ensembles, but does not process the raw ensembles themselves. The second one is a CNN which operates on some derived features, the raw ensembles and the observational data of previous days.

Both models used discrete CRPS as loss function.

4.1 The linear model

4.1.1 Data

The data is split to three separate sets for training, validation and testing. The training set contains the data of calendar year 2020, the testing set and the validation set are two distinct set from the following year. The first 60 days of 2021 are used as validation samples.

The base features are:

- the mean and standard deviation of the ensemble without the control,
- the control forecast.

The ensemble is recoded to the 10 given categories. The method of recoding is similar to the case of observations (explained in Chapter 3.3.) Then, the frequency of each category is calculated and included in the feature set.

4.1.2 Architecture

The model is a simple Multilayer Perceptron with one hidden layer containing 25 units. As activation the Rectified Linear Unit (ReLU) function is used. The output layer has equal number of units as the number of categories, its output is passed through the Softmax function to generate a probability distribution over the categories.

4.1.3 The training process

The learning rate was initialized at 0.01, then it was slowly decreased during the training. A step scheduler was used which multiplies the learning rate with a given factor γ after each 200 epochs (in this case). The ideal value for the multiplier was found out by fine-tuning.

The model was trained for 1000 epochs at maximum, but the training was stopped if a stopping criterion has been met. After the 50th epoch if in 5 consecutive epochs the difference between the actual validation loss and the mean of the last 50 epoch's validation losses is smaller than a given threshold, or the actual loss is higher than the mean, the training is stopped. The threshold used for that model was 25.

Since there are only a small set of features and all of them convey different type of information, L2 regularization was used to prevent overfitting. L2 regularization encourages the network to use small weights, but does not necessarily encourage sparsity in contrast to L1 regularization. The regularization rate was found by fine-tuning.

4.1.4 Hyperparameter fine-tuning

The ideal γ learning rate decay multiplier and the L2 regression (weight decay) factor was found in a looped hyperparameter fine-tuning process. All models showed improvement compared to the raw ensemble mean around 17.1 % on the validation set. The range of the improvements was 0.35 %, which means that these hyperparameters have almost no effect on the model's performance. Nonetheless, the best combination was used for evaluating the results, which was weight decay = 0.1 and $\gamma = 0.75$.

4.2 The convolutional model

4.2.1 Data

Similar to the linear model, the two year's data is split into three sets. The training set is derived from the data of 2020, while the validation set contains the first 60 days of 2021, the testing set contains the others.

For the convolutional model a 235 length feature set is used which comes together from 3 groups of features: base features, ensemble and historical features.

- The ensemble forecast is included for each day in raw format, without any change.
- The historical features are the observations of the previous 180 days.
- The base features include:
 - the mean and standard deviation of the ensemble (without the control),
 - the mean and standard deviation of the historical data,
 - the control forecast.

Since 180 days of data is necessary for producing the features for one day, the first 180 days of the dataset is not included in the training samples, only in the features. Additionally, a forecast for a given day with the largest leadtime (240 hours) can only make predictions based on the information which is available 10 days before the given day. Due to that, 10 extra days are left out from the training samples. The training period starts at 10-07-2020 and goes until 31-12-2020.

4.2.2 Architecture

The model consists of two main parts: a convolutional feature extractor and a fully connected classifier. The convolutional feature extractor only processes a part of the features: the ensemble and the historical data (the observations of the previous days), base features are only included in the classification.

Feature extractor

The ensemble and the historical data represent two different types of information, thus the filters required to extract the contained patterns have to be learned distinctively. Therefore, the feature extractor process them separately in two branches of convolutional layers. Both branches have the same architecture, which includes convolutional layers followed by activation functions (ReLU) and dropout layers. The dropout layers are turned off for inference.

The number of convolutional layers, the type of pooling, the kernel size (both for the convolution and the pooling) and the dropout rates are flexible attributes. The model described here uses two 1D convolutional layers per branch with stride = 1 and pads the input so the output has the same shape as the input. The pooling layers use the same stride and padding. The type of pooling is mutable - average and max pooling had been tested, the latter gave better results. The ideal combination of the other parameters were found by a looped parameter fine-tuning process.

Classification head

The output of each branch is concatenated with the base features and passed to the classifier. The classifier is built up of linear layers activated by ReLU, with dropout layers between them. The number of layers and the number of units they have are configurable. The dropout rate is also adjustable, the ideal value is found in the fine-tuning process.

4.2.3 The training process

Two phase training

The model is trained in two phases. For each test or validation day a pretrained model is actualized with a sliding window training. First, it is pretrained on year 2020 for 1000 epochs. After that, for each test or validation day (from 2021) the model is actualized for 500 epochs on 30 days preceding the given day. The effect of actualization was examined for the classification head and the feature

extractor separately. The classification head is actualized in every case, but the feature extractor can be frozen meaning that its weights are not updated during the training. Hyperparameter fine-tuning was run for both cases.

Training details

Separate models are trained for each lead time.

The same type of step scheduler was used for realizing the learning rate decay as in the Linear model. The learning rate started at 0.01 for both the pretraining and the actualization. The period of learning rate decay was 200 epochs for the pretraining, 100 epochs for the actualization. The ideal value for the multiplier γ was found in the fine-tuning process.

During the pretraining, similar to the linear model, the model was trained for 1000 epochs or until the stopping criterion had been met. The threshold was 10 in this case. During actualization, since there is no validation, the stopping criterion was evaluated on the training losses. The threshold was set to 1, since there is less, but more recent data, thus the likelihood of overfitting is lower.

For preventing overfitting dropout layers and L1 regularization were used. L1 regularization was chosen because it penalizes the absolute value of the weights in the network, therefore it encourages using sparse weights. It is best suited for a situation where there is a large number of inputs and it is rational to assume that not all of them is important. In this model a large number of historical observations is used and converted to features and it is likely that none of them alone has a great significance.

4.2.4 Hyperparameter fine-tuning

A (hyper)parameter fine-tuning loop was run for the pretraining and the actualization separately.

Pretraining

The size of the convolutional kernel, dropout rate for the convolutional and the linear layers, the rate of the L1 regularization, and the γ value for the learning rate scheduler were examined.

Attribute	Tested values
convolutional kernel size	3, 5, 7
dropout (convolution)	0.25, 0.4, 0.5
dropout (linear)	0.1, 0.25, 0.4
L1 regularization rate	0.001, 0.01, 0.05
LR scheduler γ	1, 0.75, 0.5

The best 5 combinations were chosen for further testing in the actualization phase. The best improvement was 28.6 % with $\gamma = 0.75$, kernel size = 7, dropout rate for convolutional layers = 0.4, dropout rate for linear layers = 0.1, L1 regression rate = 0.05. The average improvement was 21.6 % with range of 36.7 %, which indicates that hyperparameters have a large significance for that model.

Actualization

The best 5 pretraining combination were used to pretrain the base models. In the case when the feature extractor was frozen, dropout rate for the linear layers, the rate of the L1 regularization, and the γ value for the learning rate scheduler were examined. When the feature extractor was not frozen, dropout of the convolutional layers was also inspected.

Attribute	Tested values
convolutional kernel size	3, 5, 7
dropout (linear)	0.1, 0.25, 0.4
L1 regularization rate	0.001, 0.01, 0.05
LR scheduler γ	1, 0.75, 0.5, 0.25
<i>dropout (convolution)</i>	<i>0.1, 0.25, 0.4</i>

Freezing the convolutional layers gave same results as not freezing them, indicating that the filter weights are not modified much during the actualization process. In the case of frozen convolutional layers the maximum improvement compared to the original ensemble was 40.25%. The mean improvement was 35.52%, the range of the improvements was 9.52%.

For the unfrozen case maximum improvement compared to the original ensemble was 40.26%. The mean improvement was 36.88%, the range of the improvements was 9.34%.

Chapter 5

Results

In this chapter we show the results of the linear and the convolutional model described in Chapter 4. As reference approaches we consider the MLP and POLR models of Baran *et al.* (2021), and the raw ensemble as well. The various forecasts are evaluated on the time-period from 02-03-2021 to 31-12-2021. For comparing the forecasts skills mean CRPS and LogS over the test period and the corresponding skill scores CRPSS and LogSS, described in Chapter 2.4, are used.

The models are identified with the following notations:

- CNN-Actlzd: Pretrained convolutional model actualized with frozen convolutional layers on the 30 preceeding days of the test day (described in Chapter 4.2.3).
- CNN-Actlzd-Unfr: Pretrained convolutional model actualized with unfrozen convolutional layers on the 30 preceeding days of the test day
- CNN-Prtrn: Pretrained convolutional model without actualization
- Linear: Linear model (described in Chapter 4.1)
- POLR and MLP: models described in Baran *et al.* (2021)
- Raw: Raw ensemble

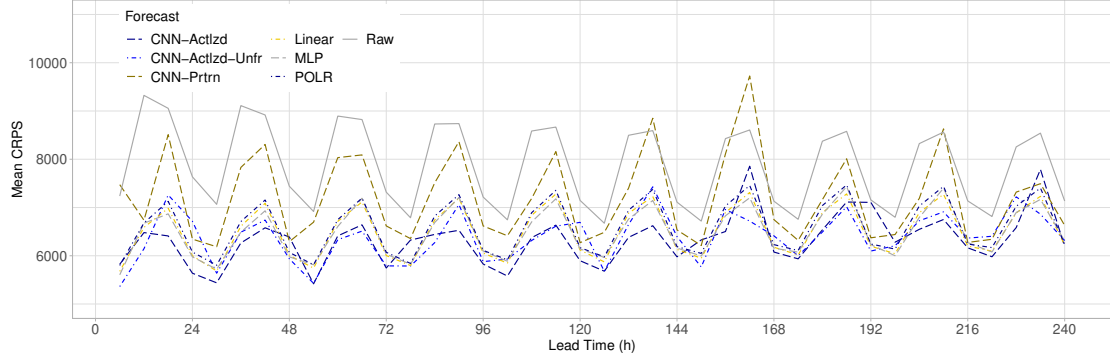


Figure 1: Mean CRPS of post-processed forecasts and raw ensemble per lead time

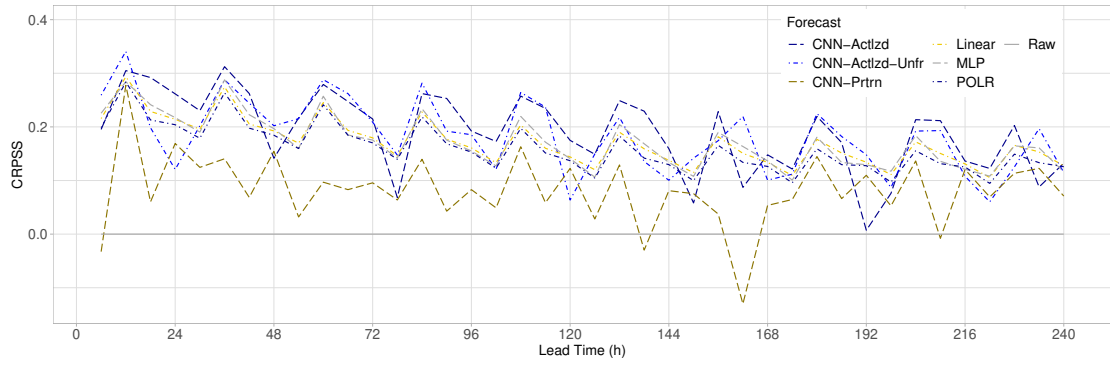


Figure 2: CRPSS of post-processed forecasts per lead time with respect to the raw ensemble

As Figures 1 and 2 indicate, on average, all models outperform the raw ensemble. The actualized model shows a mean improvement of 18.96% in terms of CRPSS compared to the raw ensemble. That is slightly more than the gains with the MLP, the Linear and the POLR models, which improved 17.18%, 17.09% and 15.94%, respectively. The CNN model without actualization only showed 8.25% improvement.

Note that the MLP and the Linear model use very similar approach to address the problem which is also reflected in the results. However, the MLP uses rolling training periods - it is trained on the preceding 350 days before the test day, while the Linear model is only trained once on the training year. This difference does not seem to have significant effects on the results.

The pretrained CNN shows little improvement compared to the raw ensemble and it seems to be very sensitive to changes in the ensemble data. However,

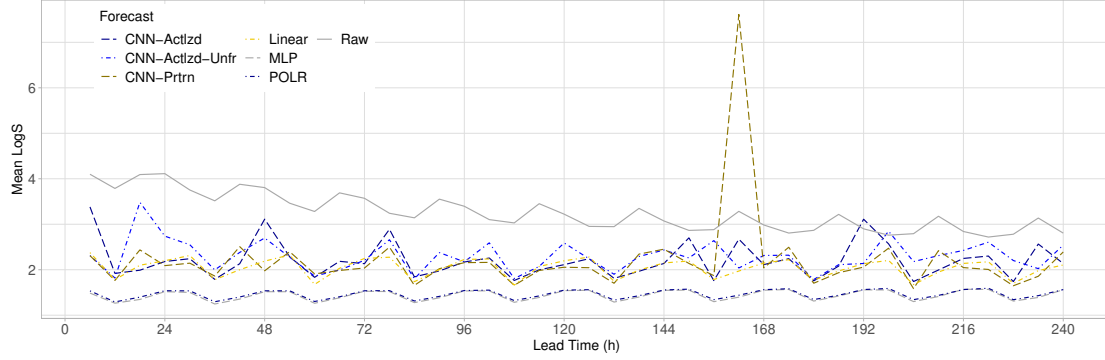


Figure 3: Mean LogS of post-processed forecasts and raw ensemble per lead time

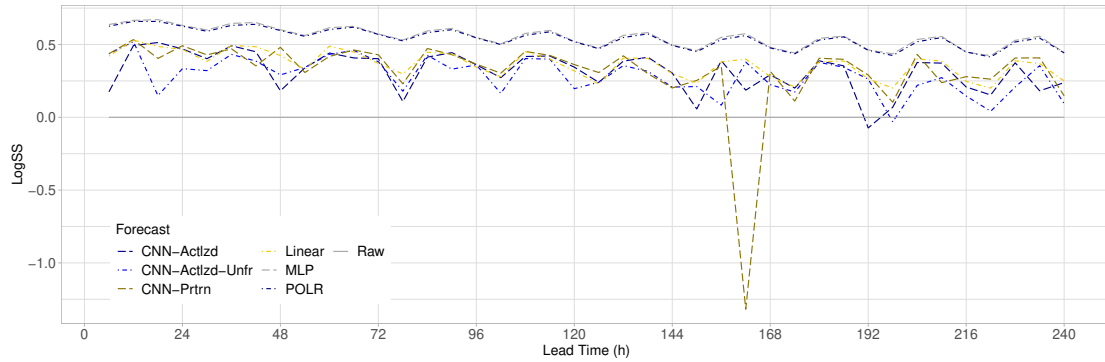


Figure 4: LogSS of post-processed forecasts per lead time with respect to the raw ensemble

actualization improves the results significantly, the CNN-Actlzd model produces the best results among the tested modes. Looking at Figure 2 it is also conspicuous that when all models show greater improvement, the CNN-Actlzd tends to outperform them. However, in some cases when all models improve less, the CNN-Actlzd shows even worse results. It can be assumed that the convolutional layers make the model capable of catching patterns in the data more sensitively than the linear models. Possible solutions for enhancing the model are discussed in Chapter 6. Optimizing the filters in the actualization process does not seem to cause significant differences in overall performance.

Figures 3 and 4 show that in terms of LogS the POLR and the MLP models give the best results. There is one conspicuous drop in the score of CNN-Prtrn at lead time 162; however, it is corrected by the actualization process.

The PIT histograms of Figures 5 and 6 show even larger differences in calibra-

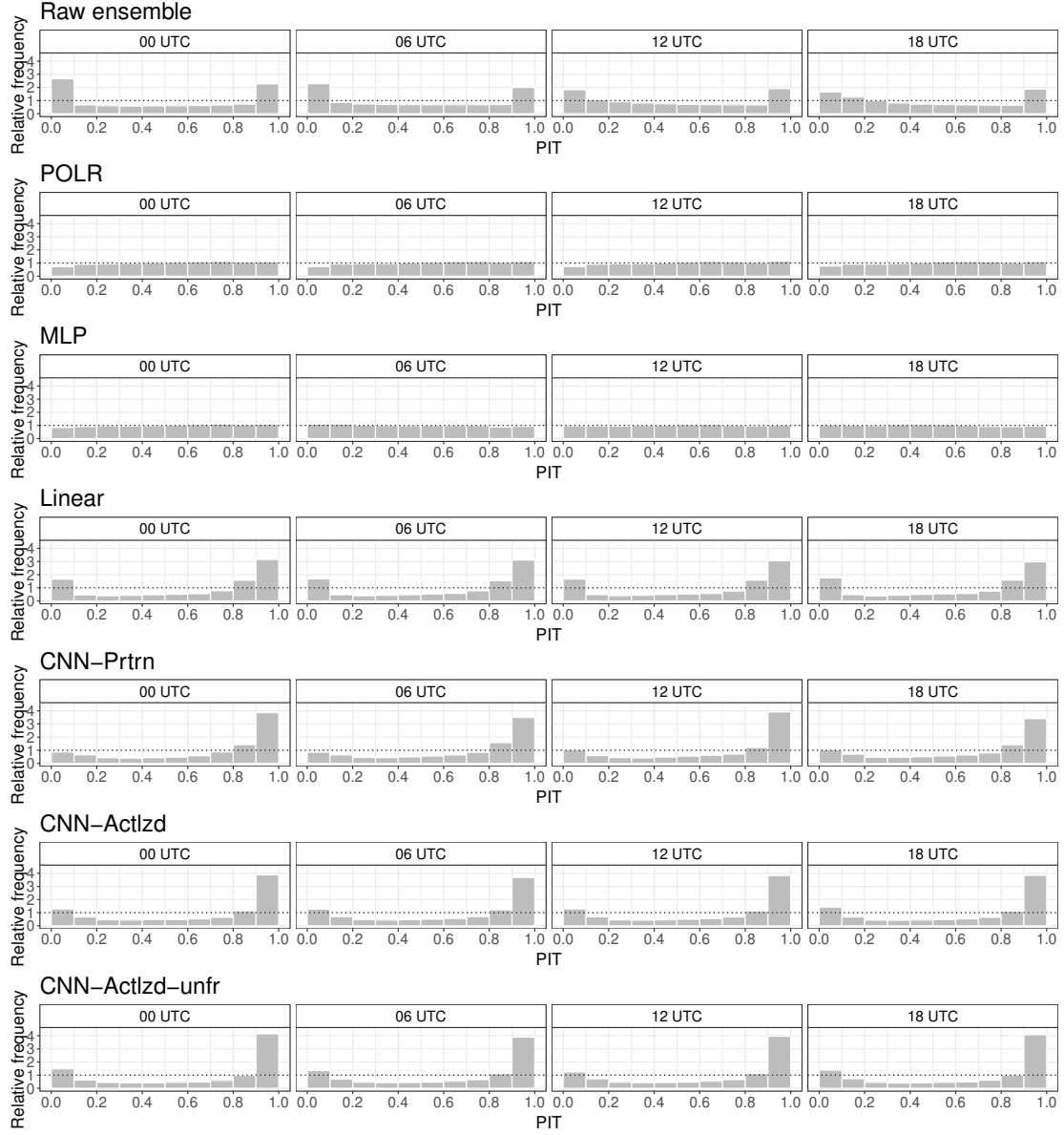


Figure 5: PIT histograms at the 4 different observation times

tion. The raw ensemble results in highly U-shaped histograms for all observation times and all lead line intervals indicating strong underdispersion. The PIT values of the MLP and POLR models follow a nearly standard uniform distribution, while the Linear and both CNN models are again U-shaped and slightly biased, which is a sign of poor calibration.

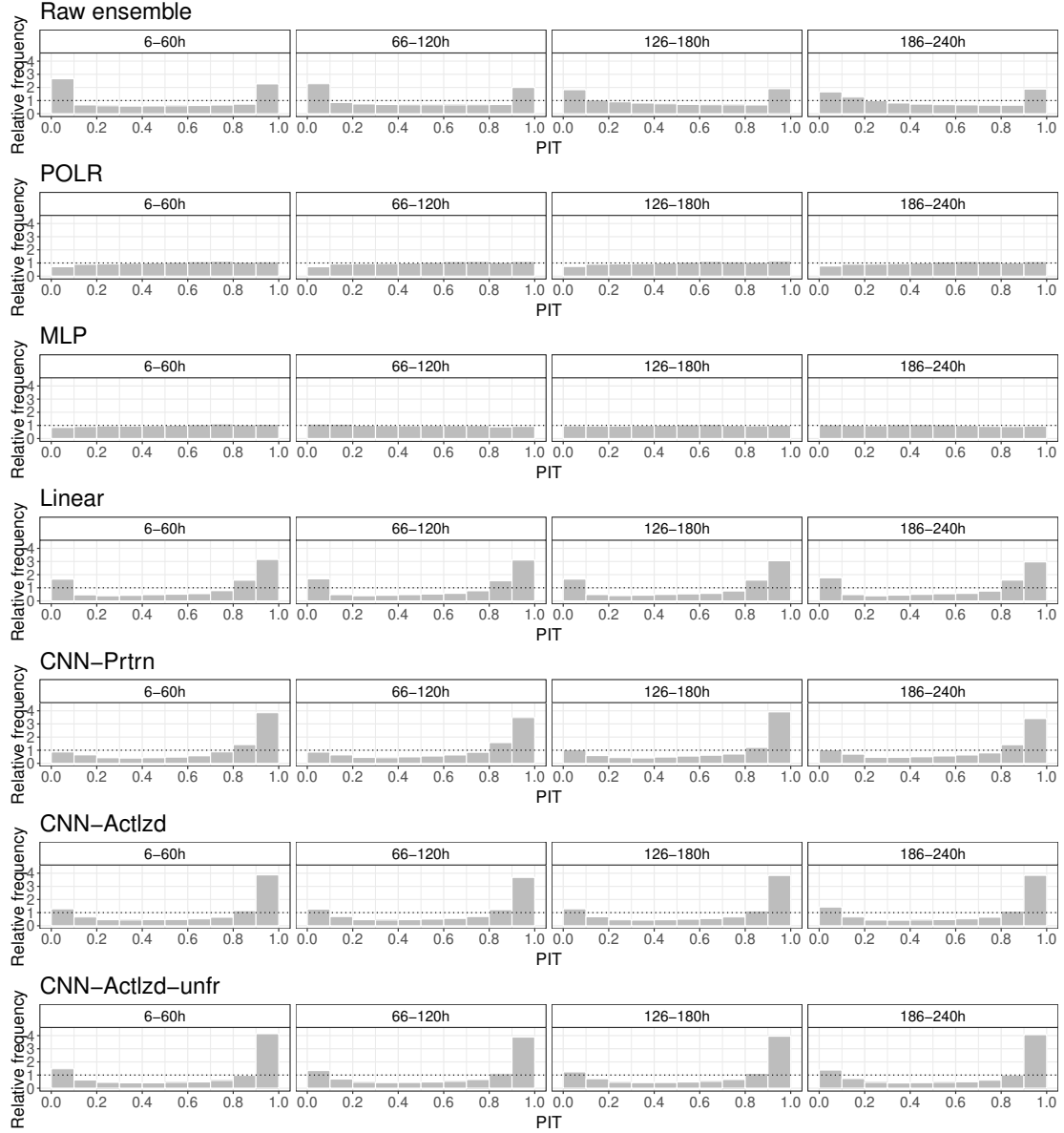


Figure 6: PIT histograms at 4 different ranges of leadtimes

Despite the improvement in CRPSS, the CNN models seem to perform worse than the other models regarding LogS or PIT. Even in terms of CRPS the result of the CNN is not stable - the CNN-Actlzd model gave the best results in most of the cases, e.g. 7.59 % better results than the second best model at lead time 90, but, in some other cases it underperforms them - with 12.69 % in the worst

case. It is also notable looking at the PIT histograms that the NN models perform worst in the edges of the range, therefore it can be assumed that these models do not handle outliers well.

Chapter 6

Discussion

Since the CNN model seems more sensitive to changes in the input data, it might be profitable to use more convolutional layers to catch higher levels of features. E.g. some minor changes in the historical data might indicate the occurrence of an unlikely event, however, the pattern might be too subtle to be recognizable with only two convolutional layers. Using the feature extractor of widely-used, state-of-the-art CNN models, like U-Net in the work Grönquist *et al.* (2021) might also be beneficial.

It is clear from the results that the NN models only perform well in the score which they were optimized for - in other terms tend to underperform the other models. The use of different loss functions (e.g. LogS) or the combination of them might be able to address this problem and result better calibrated forecasts.

The handling of outliers and preparing the model for extreme events would also be important. Training a different model for edge cases and identifying the periods when extreme events can be expected (e.g. with anomaly detection approaches) could make the model more stable.

CNN-s are designed to recognize patterns mainly in spatial data. Trying approaches that are specifically designed to process temporal data, e.g. recurrent neural network is also reasonable.

Chapter 7

Conclusion

Producing accurate and reliable forecasts are essential both for economical and safety reasons. Even state-of-the-art NWP models cannot produce perfect predictions due to uncertainty introduced by several different reasons. Ensemble forecasts address uncertainty by predicting many future states with the same NWP model with perturbed initial conditions and/or parametrization. (Buizza, 2018) A forecast ensemble often requires postprocessing to achieve proper calibration. Many postprocessing methods were developed in the last decades - completely statistical models, models that involve the usage of NNs in some parts of the process, and completely NN based models as well (Vannitsem *et al.*, 2018; Schultz and Lerch, 2022).

In this work we consider a 51-member ensemble visibility forecast provided by the ECMWF. Since visibility is reported in categories, usually following the WMO suggestions (WMO, 2014), predicting visibility can be considered as a classification problem. We developed two different models for predicting the probability distribution over 10 visibility categories from the ensemble data and the observations of previous days.

In Chapter 2 of this paper we briefly describe the sources of uncertainty, ensemble forecasting and EPSs, together with some widely used statistical and NN based postprocessing methods. Verification scores which we used to evaluate the models are also introduced in this chapter. In Chapter 3 the ECMWF visibility ensemble forecasts are described and the aspects we have taken into consideration

in choosing the 10 categories. The models and the training process are described in Chapter 4.

We developed a Linear model with two hidden layers which operates only on the ensemble data, and a CNN which makes predictions using both ensemble data and observations of the previous days. Both models were optimized for CRPS. Different parameters and hyperparameters were tried, and the effect of actualizing the pretrained CNN model on a rolling 30-day training period was also explored. All models show significant improvements compared to the raw ensemble; however, the CNN results in little upgrade over the Linear and the other tested approaches. The results are described in detail in Chapter 5.

The usage of NNs give many opportunities to improve existing postprocessing methods or create completely new approaches to address uncertainty. However, it is necessary to resolve the problem in depth and develop an architecture fitting the behaviour of the predicted weather quantity. (Similar how different probability distributions are chosen for different quantities in nonhomogenous regression methods.) Based on the consequences derived from the results we also see ways to improve our models - they are discussed in Chapter 6. Further research has to be done to develop stable models which achieve significant improvements. In the last decade NNs are gaining space in ensemble postprocessing. Hopefully they will be able to contribute to the improvement of operational EPSs in the future.

Chapter 8

Own work

The models were implemented in Python using the PyTorch framework. The models and the related scripts - for training, hyperparameter finetuning and evaluation - were developed and implemented by me. I recieved the dataset containing the ensemble data and the observations together for all leadtimes from my supervisor, Dr.Sándor Baran. The script converting the R dataset to Python compatible format, and the Python script which generates the feature matrices for both models were also my own products. Finally, I prepared the figures and performed the detailed analysis of the results.

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Dr. Baran Sándor, for his help and assistance at every stage of the work. I am especially grateful that he kept his patience even when I made the same mistakes multiple times during the experiment.

I am also deeply grateful for my family and friends who supported me emotionally. I am also thankful for my colleagues and my team leader who kept me enthusiast and let me prioritize this work over my job-related duties even when my work would have been extremely important.

Bibliography

- Baran, S. (2014) Probabilistic Wind Speed Forecasting Using Bayesian Model Averaging with Truncated Normal Components. *Computational Statistics and Data Analysis* **75**, 227–238.
- Baran, S. and Nemoda, D. (2016) Censored and shifted gamma distribution based EMOS model for probabilistic quantitative precipitation forecasting. *Environmetrics* **27**, 280–292.
- Baran, S. and Lerch, S. (2016) Mixture EMOS model for calibrating ensemble forecasts of wind speed. *Environmetrics* **27**, 116–130.
- Baran, Á., Lerch, A., El Ayari M. and Baran, S. (2021) Machine learning for total cloud cover prediction. *Neural Computing and Applications* **33**, 2605–2620.
- Baran, S. and Lakatos, M. (2023) Statistical post-processing of visibility ensemble forecasts. *Manuscript*
- Bremnes, J. B. (2004) Probabilistic forecasts of precipitation in terms of quantiles using NWP model output. *Monthly Weather Review* **132**, 338–347
- Bremnes, J. B. (2020) Ensemble postprocessing using quantile function regression based on neural networks and Bernstein polynomials. *Monthly Weather Review* **148**, 403–414
- Box, G., and Cox, D. (1964) An analysis of transformations. *Journal of the Royal Statistical Society B* **26**, 211–252.
- Buizza, R. (2018) Ensemble forecasting and the need for calibration. in S. Vannitsem, D. S. Wilks, and J. W. Messner (Eds.), *Statistical Postprocessing of Ensemble Forecasts*. Elsevier, Amsterdam.

- Christensen, H. M., Moroz, I. M. and Palmer, T. N. (2015) Stochastic and Perturbed Parameter Representations of Model Uncertainty in Convection Parameterization. *Journal of the Atmospheric Sciences* **72**, 2525–2544.
- Chmielecki, R. M. and Raftery, A. E. (2011) Probabilistic visibility forecasting using Bayesian model averaging. *Monthly Weather Review* **139**, 1626–1636.
- C. Szegedy et al. (2015) Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015* 1–9.
- European Centre for Medium-Range Weather Forecasts (2012) *ECMWF Newsletter No. 133*, ECMWF
- European Centre for Medium-Range Weather Forecasts (2021) IFS Documentation CY47R3 - Part IV Physical processes. *ECMWF, Reading*, ECMWF
- Gneiting, T., Raftery, A. E., Westveld, A. H., and Goldman, T. (2005) Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *J. Monthly Weather Review* **133** 1098–1118.
- Gneiting, T., Balabdaoui, F. and Raftery, A. E. (2007) Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society Series B* **69**, 243–268.
- Gneiting, T., Raftery, A. E., (2007) Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* **102**, 359–378.
- Good, I. J. (1952) Rational decisions. *Journal of the Royal Statistical Society Series B* **14** 107–114.
- Grönquist, P., Yao, C., Ben-Nun, T., Dryden, N., Düben, P., Li, S., and Hoefler, T. (2021) Deep Learning for Post-Processing Ensemble Weather Forecasts, *Philosophical Transactions of the Royal Society A* **379**, paper 20200092.
- Jewson, S., Brix, A. and Ziehmann, C. (2004) A new parametric model for the assessment and calibration of medium-range ensemble temperature forecasts. *Atmospheric Science Letters* **5** 96–102.
- Koenker, R. and Bassett, G. (1978) Regression Quantiles. *Econometrica* **46** 33–50.

- Leith, C. E. (1974) Theoretical skill of Monte Carlo forecasts. *Monthly Weather Review* **102**, 409–418.
- Lorenz, E. (1963) Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences* **20**, 130–141
- Lorenz, E. (1993) *The Essence of Chaos*. University of Washington Press, Seattle.
- Molteni, F., Buizza, R., Palmer, T. N., Petroliagis, T. (1996) The ECMWF ensemble prediction system: methodology and validation. *Quarterly Journal of the Royal Meteorological Society* **22**, 73–119
- Raftery, A. E., Gneiting, T., Balabdaoui, F. and Polakowski, M. (2005) Using Bayesian model averaging to calibrate forecast ensembles. *Monthly Weather Review* **133**, 1155–1174.
- Rasp, S., and Lerch, S. (2018) Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review* **146**, 3885–3900
- Ronneberger, O., Fischer, P. and Brox, T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. Available: <http://arxiv.org/abs/1505.04597>
- Roulston, M. S., Smith, L. A. (2002) Evaluating probabilistic forecasts using information theory. *Monthly Weather Review* **130**, 1653–1660.
- Scheuerer, M. (2014) Probabilistic quantitative precipitation forecasting using ensemble model output statistics. *Journal of the American Statistical Association* **140**, 1086–1096.
- Scheuerer, M. and Hamill, T. M. (2015) Statistical postprocessing of ensemble precipitation forecasts by fitting censored, shifted gamma distributions. *Monthly Weather Review* **143**, 4578–4596.
- Schulz, B., El Ayari, M., Lerch, S. and Baran, S. (2021) Post-processing numerical weather prediction ensembles for probabilistic solar irradiance forecasting. *Solar Energy* **220**, 1016–1031.

- Schultz, B. and Lerch, S. (2022) Machine learning methods for postprocessing ensemble forecasts of wind gusts: a systematic comparison. *Monthly Weather Review* **150**, 235–257.
- Sloughter, J. M., Gneiting, T. and Raftery, A. E. (2010) Probabilistic wind speed forecasting using ensembles and Bayesian model averaging. *Journal of the American Statistical Association* **105**, 25–37.
- Thorarinsdottir, T. L. and Gneiting, T. (2010). Probabilistic forecasts of wind speed: Ensemble model output statistics by using heteroscedastic censored regression. *Journal of the American Statistical Association* **173**, 371–388.
- Vannitsem, S., Wilks, D. S. and Messner, J. (Eds.) (2018) *Statistical Postprocessing of Ensemble Forecasts*. Elsevier, Amsterdam.
- Weyn, J. A., Durran, D. R. and Caruana, R. (2020) Improving Data-Driven Global Weather Prediction Using Deep Convolutional Neural Networks on a Cubed Sphere. *Journal of Advances in Modeling Earth Systems* **12**
- Wilks, D. S. (2019) *Statistical Methods in the Atmospheric Sciences, Fourth Edition*. Elsevier, Amsterdam.
- World Meteorological Organization (1966) *International Meteorological Vocabulary* (WMO-No. 182). WMO, Geneva.
- World Meteorological Organization (2010) *Manual on the Global Observing System* (WMO-No. 544 Volume I.). WMO, Geneva.
- World Meteorological Organization (2012) *Guidelines on Ensemble Prediction Systems and Forecasting* (WMO-No. 1091). WMO, Geneva.
- World Meteorological Organization (2014) *Measurement of Meteorological Variables* (WMO, 2014 Edition, Chapter 9). WMO, Geneva.
- World Meteorological Organization (2021) *Guidelines on Ensemble Prediction System Postprocessing* (WMO-No. 1254). WMO, Geneva.