Deep learning methods for forecasting of extreme ambient ozone values

Vincent Gramlich

2.12.2021

Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Thesis reviser:
Prof. Dr. Melanie Schmidt (Universität zu Köln)
PD Dr. Martin Schultz (Forschungszentrum Jülich)

Abstract

Exposure to high ozone concentrations can be harmful for humans and therefore many countries have declared a threshold for ozone concentrations that should not be exceeded. That is why the ability to predict high and extreme near surface ozone concentrations is important not only for human health but also for regulatory purposes. The problem with many existing ozone forecasting methods, especially deep learning approaches, is their inaccuracy and unreliability to forecast high ozone concentrations. The goal of this study is to discover the usage of oversampling and subsequent finetuning to increase the forecast precision for extreme near surface ozone concentration. Therefor the architecture and experiment setup of IntelliO3-ts, a convolutional neural network for the forecast of near surface ozone concentrations, is used as a foundation to which the methods are applied. At first, oversampling is applied to the data set, which is the process of multiplying samples from less frequent ozone concentration ranges and adding them to the data set. The thereby obtained new "oversampled" data set that has a flatter sample distribution is then used to train the neural network. In a second and additional step the finetuning takes place, which is a retraining of the network obtained in the first step, using the original data set before oversampling was applied. For both methods different parameters will be tested and evaluated on the basis of different scores calculated on 2×2 contingency tables. The contingency tables are created by using a threshold and separating the test data in two groups, ozone concentrations below and above the threshold.

The oversampling increases the ability to successfully forecast if a sample exceeds a certain threshold, thereby increasing the forecast precision for high ozone values. These advantages come at the cost of also increasing the percentage of samples that are falsely predicted to be above a certain threshold, also resulting in a systematic overestimation. The best model obtained, was able to increase the hit rate at 60 ppb from 43% to 67% and at 80 ppb from 1.9% to 15.2%. This means that the model is able to correctly predict that a sample is above 60 ppb and 80 ppb for 67% and 15.2% of all samples above that threshold instead of only achieving this for 43% and 1.9% of the samples above that threshold, respectively. Therefore the oversampling offers a valuable trade off, to sacrifice parts of the overall performance in order to increase the ability to forecast high ozone values, which might be useful especially for regulatory purposes. The finetuning did not add any new value to that, but only reverted some of the improvements that were achieved by the oversampling.

For future research it might be interesting to investigate the usage of different oversampling methods or explore the application of oversampling and finetuning to the forecasting of multiple days, as this study only focused on the forecast for the next day.

Contents

1 Introduction

1.1 Ozone and its emergence

Ozone is a highly reactive gas which is present in the stratosphere and the troposphere [50]. The ozone in the stratosphere forms the ozone layer which protects life on earth from high doses of harmful UV radiation, but the ozone in the troposphere can be harmful for humans, animals and crops [53][39]. Especially high concentrations of tropospheric ozone can cause respiratory problems, decrease physical capabilities and worsen many other diseases. The German "Umweltbundesamt" defines the critical threshold for the 8-hour average ozone concentration as $120\mu/m^3$. This critical threshold should only be exceeded on 25 days in a year [49]. This fixed threshold is not unique to Germany and shows the relevance of the ability to forecast high or "extreme" ozone concentrations. It is not only directly important for human health but it also gives regulators and government authorities the ability to evaluate goals beforehand and to react and adapt in advance.
Unlike other air pollutants ozone is not directly emitted into the atmosphere but arises from various reactions of primary air pollutants. Hence, ozone is classified as a secondary air pollutant. The main primary air pollutants that are responsible for the emergence of ozone are hydrocarbons (CH_x), volatile organic compounds (VOCs) and nitrogen oxides (NO_x). The NO_x are broken apart by UV radiation in the sunlight and the free radical oxigen (O_1) bonds to dioxygen (O_2) to form ozone (O_3) [46]. In this process, temperature is an important factor but in general also other weather conditions have a major influence on ozone formation and destruction. Since the different air pollutants and weather conditions also influence each others, the dependencies for the emergence of ozone are highly non linear. The non-linearity of these dependencies and the time delay with which they take place makes the forecast of ozone a complex task. Fortunately across Germany as well as many other countries there is a dense web of air quality measuring stations which provides huge amounts of data for various air pollutants [48] and the COSMO-REA6 provides reanalysis for meterological data [35], both are accessible via the TOAR database [43].

1.2 Climatological Forecasts

Ozone forecasts, just like other forecasting tasks, can be divided into two groups, statistical methods and numerical models. Numerical models for weather and air quality forecasts emerged in the beginning of the last century. These methods use various numerical schemes to solve chemical transport models [11]. Today numerical models are still used among a broad range of air quality and weather forecasting tasks [27]. Statistical methods have a long history and include simple persistence forecasts or more complex regression models [40] [2]. With the success of machine and especially deep learning across various research fields, a new and vastly growing field of statistical methods emerged. This new development was also picked up by a growing number of researchers that study enviromental forecasting. Comrie et al. were one of the first researchers to investigate this possible application [12]. They used the first existing neural network architecture called fully connected networks and many others followed their example [10][51]. Since then, many of the more complex methods and architecture that emerged from other deep learning applications have been applied to enviromental forecasting [31]. Tai-Long et al. [21] among many others [4][33] explored the usage of recurrent neural networks (RNNs), an architecture which was specifically designed for sequential data. Convolutional neural networks (CNNs), which were conceptualized for the extraction of different features and initially used in computer vision, where used for ozone forecasts by Sayeed et al. [41] or Eslami et al. [17]. A more detailed description of the different neural network architectures is provided in section 2.

1.3 Extreme (Ozone) Values and their Forecasting

Extreme concentrations of near surface ozone are generally often defined as concentrations that exceed a certain threshold or more general as a strong deviation from average concentrations. In Germany the threshold for the 8-hour average ozone concentration is at $120\mu/m^3$ and in most regions and years this threshold is only exceeded 1-10 times per year [49][47]. So although there are plenty of air quality measuring stations, many of them with records lasting 10 years or more, there are still very few samples for extreme ozone concentrations. This leads to the question whether these few samples are sufficient for deep learning methods to learn from them in order to predict future extreme ozone concentrations. While there have been efforts to specifically target the forecasting of extreme ozone values with chemical transport models [42], the approaches using pure deep learning to forecast extreme ozone values are very rare.

In general the deep learning research on extreme value forecasting is very shallow but plenty of research has been done on a similar task, the forecasting of extreme events. Those extreme events can be extreme traffic loads like in the study by Laptev et al. at Uber [36] but can also be related to climatology like extreme weather events in the study by Liu et al. [32]. These forecasting tasks are often referred to as imbalanced data problems since the frequency of the different classes, extreme and non extreme events, is very different. Buda et al. did a systematic study on imbalanced data problems and methods to tackle these [8] which was afterwards picked up and confirmed by Johnson et al. [22]. They found out that a form of data augmentation, especially oversampling which is the process of copying samples of less frequent classes, can improve forecast precision for less frequent classes. Additionally finetuning, a form of retraining of the network with the original data set, brought further improvements in their study. This raises the question whether their results and the methods that brought the best results for them are also transferable to forecasting of extreme values, specifically extreme ozone concentrations. The theoretic concept of both methods, oversampling and finetuning, will be explained in section 3 and their concrete application will be described in section 4.

1.4 Foundation and Aim of the Study

The main foundation for this study is the IntelliO3-ts paper by Kleinert et al. [26]. IntelliO3-ts is a convolutional neural network consisting of two inception blocks and trained on data from 334 monitoring stations in Germany. This study uses the same network architecture and data and aims on improving the networks ability to precisely forecast extreme values. The implementation is made with MLAir, a programming framework for machine learning on air data time series which was developed by Leufen et al. [30], and the data is accessed via the TOAR (Tropospheric Ozone Assessment Report) database [43].

The aim of this study is to increase the performance of the network on high or “extreme“ ozone values. Therefore, oversampling and oversampling with subsequent finetuning will be applied in order to investigate whether they show the same improvements for forecasting of extreme values as they did for forecasting of extreme events. In order to evaluate the effect of these methods, evaluation metrics based on 2 × 2 contingency tables with gliding thresholds are introduced.

This study is structured as followed: Section 2 will give an introduction to neural networks, their fundamental features and the different network architectures. In section 3 the characteristics of extreme ozone values will be presented and an overview on deep learning methods for extreme values and extreme events forecasting will be given. Afterwards the two methods used in this study, oversampling and finetuning, and their theoretic concepts and reasoning will be introduced. Section 4 gives an overview on the data and methods used in this study, the concrete application of oversampling and finetuning and the evaluation metrics used in this study. After that, the results of the study will be presented in section 5 and discussed in section 6. Finally in section 7, a conclusion is drawn and an outlook on potential further research questions is given.

2 Artificial Neural Networks

Artificial neural networks (ANNs) are a computer scientific and statistical tool which is able to capture highly non-linear and unknown relations in data. This ability is used for various tasks like image recognition, natural language processing or multivariate regression as done in this study. ANNs are build to mimic the structure of the human brain by processing information through different layers of “neurons“ and afterwards “learning“ from the errors made. The idea of a “logical calculus“ to imitate “nervous activity“ was conceptualized by McCulloh et al. in 1943 [34] and Rosenblatt et al. later proposed the first feed forward neural network as multilayer perceptron (MLP) [18].

After the MLPs, also called feed forward or fully connected networks, many other network architectures emerged. Two of the first new architectures were recurrent neural networks (RNNs), which were first proposed by Rumelhart et al. in 1986 [13], and convolutional neural networks (CNNs), proposed by LeCun et al. in 1998 [28]. While RNNs are specialised on sequential data by recurrently processing inputs and producing outputs, the CNNs focus on feature extraction with different convolutional filters, a data compression method that originated in image processing. Over time many more complex architectures, which are often more specialised in terms of applications or specific structures of in- and outputs, were developed. Important examples are general adversarial networks (GANs) proposed by Goodfellow et al. [19] or variatonal auto-encoders (VAEs) proposed by Kingma et al. [25] along many others. Furthermore, more detailed methods and structures were developed to enhance existing architectures. An important example for this study are inception blocks, proposed by Szedegy et al.[45], which are a specific building block for CNNs that combine multiple convolutions that are done in parallel.

This chapter contains an overview on feed forward networks and the calculations and formulas needed for their learning and forecasting, which is mainly based on [5] and [3]. Afterwards introductions to convolutional neural networks and inception blocks, which represent the main architecture used in this study, are given.

2.1 Feed forward networks

Feed forward networks, first introduced as multilayer perceptron, consist of computational neurons stacked together as layers of neurons. These layers can be divided in three different categories, input-, hidden- and outputlayers. The general task of any neural network is to learn a target function t that maps the input x to an output $\hat{y}$. The target function can be described by the following process: The initial input is fed into the inputlayer, gets passed through the hiddenlayers until it reaches the outputlayer which transforms the information into the desired output shape. This process of calculating the output $\hat{y}$ from an input x is called forward propagation and is the reason for the name "feed forward network" because the input is "fed forward", passing each layer consecutively. The target function itself is learned during a second process called backward propagation. Backward propagation is done on a training data set where for every input x a desired output y, also called label, is given. The target function first maps the input x to an output $\hat{y}$ and is than adjusted in dependency of the deviation from the prediction $\hat{y}$ to the label y.

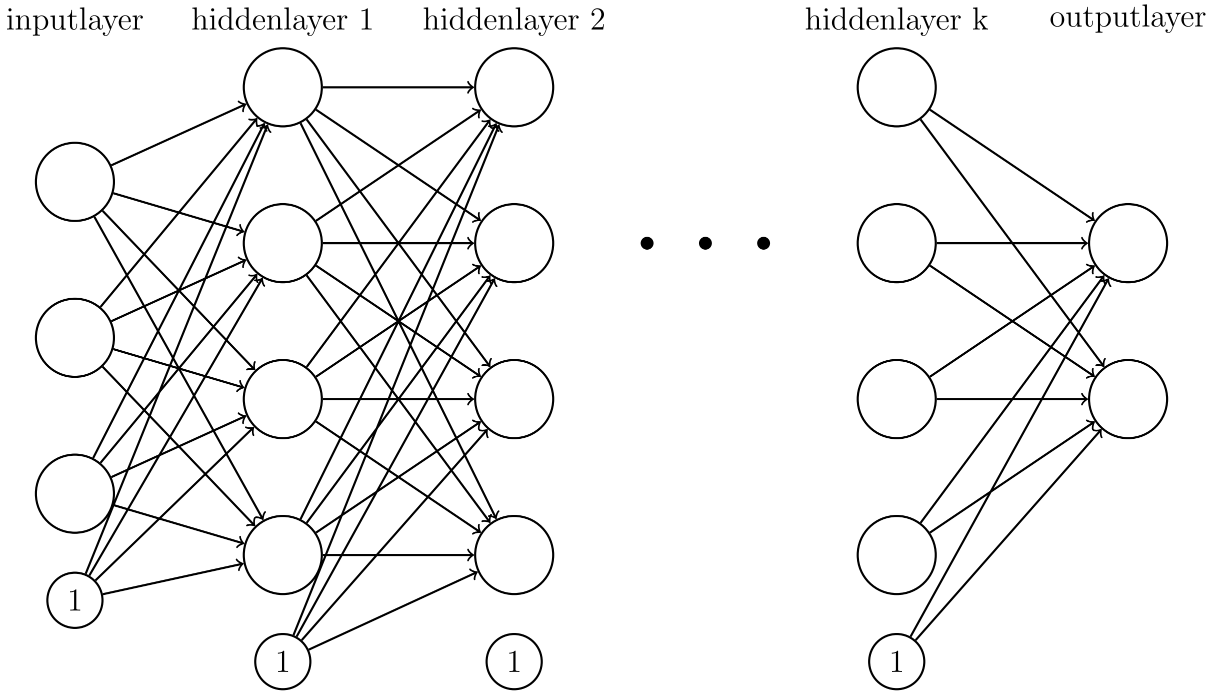


Figure 1: k-hiddenlayer neural network consisting of an inputlayer, k hiddenlayers and an outputlayer. The empty circles represent neurons that are grouped into the different layers. The small circles with 1 are used to represent the biases in the consecutive layer. The arrows between the neurons represent the links.

In total there are three main components to a neural network as can be seen in figure 1, the different layers, the neurons which make up the layers and the links that connect neurons of consecutive layers. The name fully connected network comes from the fact, that in this type of network every neuron of one layer has a link to every neuron in the consecutive layer.

The numbers that are involved in the calculations in a feed forward network can be divided into two parts, values which are calculated during the forward propagation and parameters which build the target function and are changed in the backward propagation. Each link has a parameter w which is called weight and each neuron has a parameter b, the bias. For computational reasons the bias is represented as a link to a node in the previous layer with the value 1. Each neuron contains two different values, the net input z and the nodes output a. The input value z is derived as the weighted sum of the output values a from the nodes of the previous layer, multiplied with the weights of the corresponding links plus the bias of the neuron. The value a is the output of the activation function g given z as input. For the neurons in the l-th layer the following applies:

$$\begin{aligned} z^{[l]} &= W^{[l]} * a^{[l-1]} + b^{[l]} \\ a^{[l]} &= g^{[l]}(z^{[l]}) \end{aligned} \tag{1}$$

where $z^{[l]}$ and $a^{[l]}$ are in $\mathbb{R}^{n^{[l]}}$, $W^{[l]} \in \mathbb{R}^{n^{[l]} \times n^{[l-1]}}$ is the weight matrix of the l-th layer, $b^{[l]} \in \mathbb{R}^{n^{[l]}}$ is the bias vector of the l-th layer, $g^{[l]}$ is the activation function of the l-th layer and $n^{[l]}$ is the number of neurons in the l-th layer. To save computational cost the calculations of $z^{[l]}$ is actually done as:

$$z^{[l]} = \tilde{W}^{[l]} * \tilde{a}^{[l-1]} \tag{2}$$

where $\tilde{W}^{[l]}$ consists of the weight matrix $W^{[l]}$ and additionally as last (or first) column, the bias vector $b^{[l]}$ is added. Analogously vector $\tilde{a}^{[l-1]}$ consists of the neuron values of the previous layer $a^{[l-1]}$ and an additional 1 as last (or first) entry to represent the bias neuron. This computational concept is also displayed in figure 1 where the bias for every neuron is replaced by a link to a neuron with the value 1. The bias of the neuron then is identical to the weight of this link.

The most common case for activation functions is that for all layers except the output layer a fast computable, non-linear function is used, like rectified linear units (RELU), exponential linear unit (ELU) or tanh. The non-linearity is essential, because with a linear activation function for all layers, the target function that the neural network represents would always stay linear, no matter how the different parameters are changed. For the output layer the most common choices are softmax or linear activation functions [44]. The calculations are done iteratively for every layer until the outputlayer is reached. The output layer then outputs $\hat{y}$ which completes the forward propagation.

Afterwards, if the network is being trained, the backward propagation starts where the network learns from the errors made by adjusting weights and biases in the different layers. The first step hereby is computing the loss function L from the prediction $\hat{y}$ and the correct label y. The most commonly used loss function is the mean squared error:

$$L(\hat{y}, y) = \sum_{i=1}^{n_o} (\hat{y}_i - y_i)^2 \tag{3}$$

where n_o is the size of the output layer. From this loss L, partial derivatives of every parameter in the network are calculated. Then the concept of gradient descent is applied, which means the weights and biases are changed in the direction of the negative partial derivative. The amount of change is determined by the amount of the partial derivative and the globally set learning rate α. So every parameter p, with p being a weight or a bias, is updated as:

$$p = p - \alpha \frac{\partial L}{\partial x} \tag{4}$$

For a more detailed explanation on backpropagation see [55].

2.2 Convolutional Neural Networks

Convolutional neural networks are a specific type of neural network architecture that got his name from using convolution operations, a data comprehension method that is mostly used in image processing. It was proposed by LeCun et al. [28] and initially conceptualized for the recognition of handwritten digits or the broader field of image recognition and classification. This introduction to CNNs is mainly based on the genesis paper from LeCun et al. [28] and a comprehensive but deeper insight is given by Albawi et al. [1]. All matrix indices in formulas will start at zero.

A convolutional neural network has the same main function as every feed forward network, which is to learn a target function t that maps an input x onto an output $\hat{y}$. But instead of only using layers of fully connected neurons, like a fully connected network does, some layers instead consist of one or more convolutional filters. The terms convolutional filter and kernel often have slightly different meanings but in this work, they are treated as synonyms that denote a single convolution matrix. A single filter is represented by a matrix with certain shape and values. Most commonly the filter is a quadratic two dimensional matrix which is applied to a two or three dimensional input matrix. For example consider a 8×8 input matrix M and a 3×3 filter F. The filter is applied to every 3×3 submatrix by multiplying the values of M and F at the same position and adding them up. Hence, for the output matrix O holds:

$$O_{i,j} = \sum_{k=0}^{2} \sum_{l=0}^{2} M_{i+k,j+l} * F_{k,l} \tag{5}$$

There are in total 6×6 potential positions in which the filter can be applied to the input matrix, that is why $O \in \mathbb{R}^{6\times6}$. In general for a $m \times m$ input matrix M and a $f \times f$ sized filter, the output matrix O is of shape $(m + 1 - f) \times (m + 1 - f)$ and defined as:

$$O_{i,j} = \sum_{k=0}^{f-1} \sum_{l=0}^{f-1} M_{i+k,j+l} * F_{k,l} \tag{6}$$

There are two more parameters to convolutions which are stride and padding. Stride describes the step size in which the filter is applied to the input matrix. Padding describes a process where the input matrix size is increased by adding additional rows on the top and bottom and columns at the left and right of the matrix. This ensures that the output matrix has the same shape as the input matrix before the padding was applied. The additional rows and columns are most often filled with either zeros or mirror the values in the original input matrix. In the formulas above, a stride of one and no padding is used. If in general a stride of s and a padding of p is applied, the input matrix is padded to a matrix $\tilde{M}$ which has the size $(m+2p) \times (m+2p)$ and the formula for O changes to:

$$O_{i,j} = \sum_{k=0}^{f-1} \sum_{l=0}^{f-1} \tilde{M}_{i*s+k,j*s+l} * F_{k,l} \tag{7}$$

with the number of rows and columns in O being $o = \lceil \frac{m+2p+1-k}{s} \rceil$. This means a bigger stride leads to a smaller output matrix and a bigger padding on the other hand leads to a bigger output matrix. For the output matrix to be the same shape as the input matrix, the following equation must be valid:

$$\begin{aligned} o = \lceil \frac{m+2p+1-k}{s} \rceil &\overset{!}{=} m \\ \Leftrightarrow p = \lfloor \frac{m(s-1)+k-1}{2} \rfloor & \end{aligned} \tag{8}$$

The last method which is also used in CNNs are poolings, mainly average and maximum pooling. They work in a similar way to convolutions but instead of applying a filter to every submatrix, the average or maximum value is chosen from each submatrix.

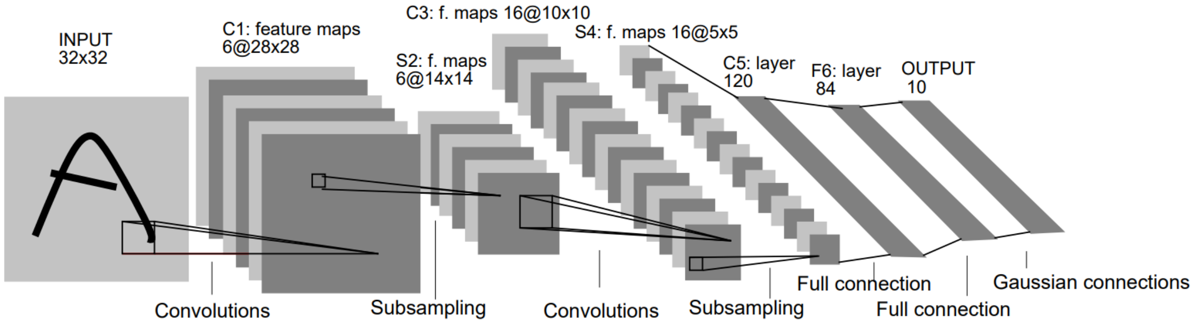
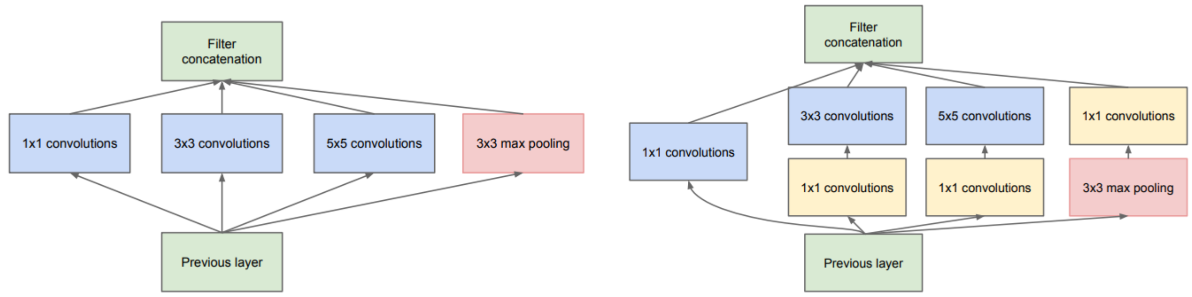


Figure 2: Illustration of LeNet-5 architecture from the paper by LeCun et al. [28] which was conceptualized to recognize handwritten digits from a 32×32 pixel input image. The image displays the shapes of the data that is outputted after every layer, the layers that are between these outputs are described at the bottom of the image. The layer 1 and 3 consist of 6 and 16 convolutional filters with the size 5×5 with no stride or padding which is why both layers reduce the input size by 4 in each dimension. The so called subsampling layer at positions 2 and 4 are 2×2 convolutional filters with a stride of 2 which is why they reduce the input size by a factor of 2. In the end 2 fully connected layer with 120 and 84 neurons are added as well as a gaussian connection that produces the output of size 10.

A CNN typically consists of one or multiple convolutional layers, often directly followed by a pooling layer. At the end of these layers a few fully connected layers complete the network. The idea of this architecture is that the convolutional layers extract the features from the input and afterwards the fully connected layers learn to combine these features. A convolutional layer consists of multiple convolutional filters. Just like the weights and biases in the classical feedforward network the weights in these filters are considered as parameters and are tuned in the backward propagation. Deeper into the network the number of filters used per layer tends to grow. Additionally the size of the input matrix gets smaller with every convolutions without padding that is applied. In the simple example of a two dimensional input matrix this leads to matrizes that are getting smaller in those two dimensions with every layer but growing in the third dimension with more and more filters used. Conceptually the first convolutional layers start extracting a few bigger features and with every extra layer the features extracted start to get smaller and more specific and their quantity grows. This concept can be observed in figure 2 which displays the architecture of the LeNet-5 network from the paper by LeCun et al. [28].

2.3 Inception Blocks

Inception blocks are a specific type of CNN architecture first proposed by Szegedy et al. in 2014 [45]. Up to this point the arising problem with convolutional neural networks was that in order to improve their performance the most commonly used approach was to make them bigger, thereby increasing the number of convolutional layers as well as the number of filters used in each layer. This expands the number of total parameters that are trainable which can make the network more powerful on the one hand but more prone to overfitting on the other hand, not to mention the exponentially rising computational costs. The second problem with this approach are vanishing or exploding gradients which refers to the partial derivative calculation in the learning process of the network. As more layers get stacked on top of each other, the risk of the gradients either converging to zero or infinity grows.



(a) Inception module, naïve version

(b) Inception module with dimension reductions

Figure 3: Illustration of the inception module from the paper by Szegedy et al. [45]. The naive version on the left side simply does 1×1, 3×3 and 5×5 convolutions and a 3×3 max pooling in parallel and concatenates these. For the inception module with dimension reduction on the right, a 1×1 convolution is added to every of the parallel processes that did not have it yet. Before the 3×3 and 5×5 convolutions and after the max pooling.

To solve theses problems Szegedy et al. designed the inception blocks which consist of different convolutions with different filter sizes and poolings that are done in parallel and concatenated at the end. The different filter sizes, pooling types and also number of convolutions and poolings used can be varied across different applications. Two examples for combinations of convolutions and poolings in a inception block can be seen in figure 3, which is taken from the paper by Szegedy et al. [45].
The network architecture used in this study consists of two consecutive inception blocks that are very similar to the inception module with dimension reductions in figure 3. Further detail will be given in section 4.

3 Task of Predicting Extreme Values

The definition of extreme values can differ between research fields but extreme values are commonly defined as data points that have an extreme deviation from the average value and therefore lie on the edge of the distribution. Prediction of extreme values is similar but not identical to predicting extreme events. Both, extreme values and extreme events, often have a low frequency but the forecasting of extreme values is a multivariate regression task while the prediction of extreme events is a classification task. This chapter will go over what characterizes extreme ozone values and which of these characteristics makes them hard to predict. Furthermore it will be described why especially deep learning methods have problems with predicting extreme values and what approaches have been developed to tackle extreme value and extreme event forecasting. Finally the theoretical concept and reasoning behind the two methods which will be investigated in this study, oversampling and finetuning, will be explained.

3.1 Extreme Ozone Values

Near surface ozone concentrations can vary strongly as can be seen in figure 4 which displays the histogram for the training data used in this study, that was standardized to mean zero and unit variance. The data distribution shows a deviation from the mean of up to six standard deviations on the right side and up to two standard deviations to the left side. This asymmetry is due to the fact, that ozone concentrations are on the lower end limited by zero which is why their distribution is called a zero-limited asymmetrical distribution. Because of this asymmetry on the right and left end of the distribution, it differs from a regular gaussian distribution. This imposes a problem since many forecasting methods implicitly assume a gaussian distribution or at least achieve the best performances on gaussian distributions. This and and the fact that extreme ozone concentrations have such a low frequency makes it difficult to predict them.

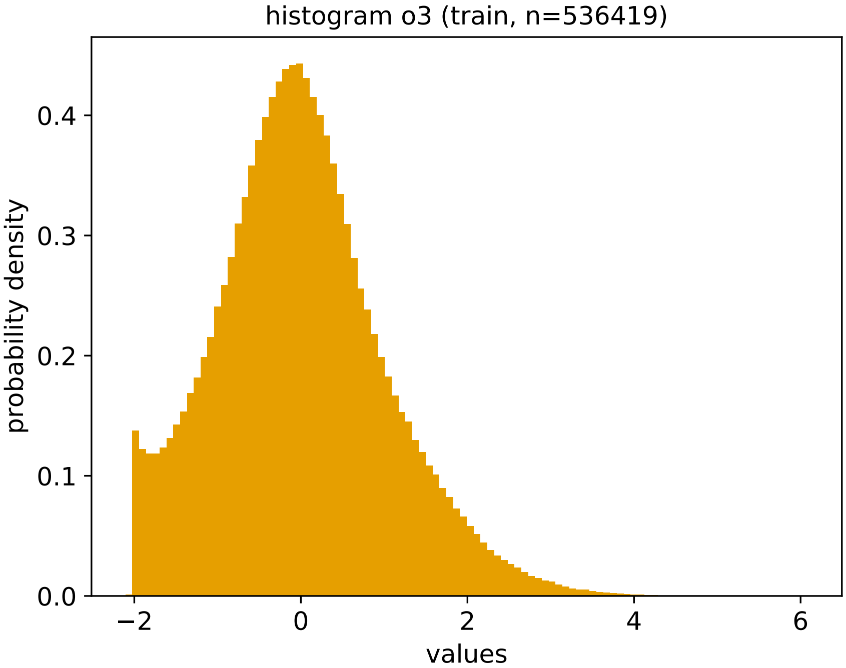


Figure 4: Density histogram of the ozone concentrations in the training data which consists of 536419 samples. The training data was normalized standardized to mean zero and unit variance and therefore the values on the x-axis are giving as standard deviation.

3.2 Extreme values in deep learning

The forecast of extreme values, especially if they are extremely rare, is a difficult task for many methods, but for deep learning this problem is especially significant. The main concept of deep learning, which is to “learn“ from large amounts of data, already highlights the two main issues. The first lies in the data that has a large amount of total samples but the extreme values only make up a small amount of that. The second problem comes with the “learning“ process of the network. The learning is based on the error of a loss function and the loss is calculated over all samples, most often with a mean squared or mean absolute error. This results in the networks prediction being drawn to the average value and especially being incentivised not to predict extreme values. Approaches to solve the latter problem often include the development of different loss functions. Examples for this approach are studies from Di Qi et al. who applied a relative entropy loss [14] or the extreme value loss proposed by Ding et al. [15]. Both are based on the cross entropy loss, a loss function that does not consider small deviations between forecast and observation but rather accounts for if the statistical structure is captured, for example detecting whether it is an extreme event or not. A key feature of the cross entropy loss and its variations is the usage of a log function which allows to ignore smaller deviations and accounts stronger for bigger deviations. The logarithmic function furthermore can serve the purpose of transforming the distribution function to an approximate gaussian distribution.

Tackling the first problem, the low frequency of outliers, is more common in the closely related field of predicting extreme events. Tasks in this field are also often referred to imbalanced data problems because of the imbalance in the distribution of the extreme and non extreme events or classes. The most common methods used can be divided into two groups, data driven and classifier based methods. Data driven methods include some form of data augmentation, mostly either increasing the number of samples for lower frequent classes or decreasing the number of higher frequent classes. This is called over- or undersampling. Classifier based methods change properties of the classifier itself by for example changing thresholds that are used to divide classes or change weights that are used for classification. Buda et al. did a systematic study on imbalanced data classification for convolutional neural networks [8] and their work was picked up by Johnson et al. in their survey on deep learning with class imbalance [22] which showed similar results. Buda et al. investigated both data driven as well as classifier based methods and evaluated these methods on three datasets with different sizes and levels of class imbalance. In their study oversampling and oversampling with subsequent finetuning, also denoted as two-phase approach, showed the best results on all datasets. Both methods will be further explained in the following. This study investigates whether the results that Buda et al. showed for the imbalanced data classification are also transferable to the prediction of extreme ozone concentrations.

3.3 Oversampling

Oversampling in general describes the process of creating more samples of a specific type which can be a class, a single value or a range of values. Most often it aims on multiplying the number of samples of a type that is less frequent than other types, thus making the data distribution more balanced. Oversampling often brings one big problem, which is overfitting. Overfitting is the process where a neural network, especially for deep neural networks, tailors itself to strong to the training data and is therefore not able to capture interrelations within the data but rather just “memorizes“ all training samples. This typically occurs when the ratio of the parameters in the network to the numbers of samples in the training set gets to high or the network is training for to long [29]. Oversampling can cause a special form of overfitting where the networks ability to detect and classify lower frequent classes is only learned on a few different samples that just occur multiple times in the training set. The network then learns to correctly classify those exact samples without learning the underlying structure which is responsible for a sample to be in this specific, low frequent class. But according to the results from Buda et al. [8] and Johnson et al. [22], convolutional neural networks are less vulnarable to overfitting as other deep learning architectures.

Oversampling methods can be divided into two main groups, duplicating or synthesizing. The latter create new “synthesized“ samples and have a broad range on how the synthesizing takes place. One of the more complex examples is the synthetic minority oversampling technique (SMOTE) [9]. SMOTE synthesizes samples by using the point representations of samples in the feature space. So for every feature the samples have, there is one dimension in the feature spaces and every sample has a coordinate according to the feature values of that sample. In every step SMOTE then takes two random samples out of the class it wants to oversample and creates a new sample that lies on a random position on the path between the two chosen samples. Which samples to choose or where on the path to create the new sample can also be decided according to different rules and heuristics. The duplicating oversampling method which is also used in this study is simpler. It just takes samples of the specific type or group and multiplies them, resulting in a dataset that contains duplicates but is more balanced. The variation in duplicating oversampling methods is only in the way the samples are chosen. In this study they are just randomly chosen from the whole group which is to be oversampled, but it is also possible to use different heuristics, for example only choosing specific subgroups of the main group that is to be oversampled.

3.4 Finetuning

Finetuning also called two-phase approach is a form of transfer learning. The concept of transfer learning is used for domains with very limited data for this very specific domain but with plenty of data for a related domain. Transfer learning systems are trained on the domain with lots of data and the gained knowledge is then "transferred" to the task with limited data by retraining (parts of) the system with this smaller data set [52]. This concept is most often applied to convolutional neural networks. For example breast cancer detection, a domain with very limited amounts of data, especially with cancer positive labeled data, can be done by transfering the knowledge of big image classification networks like GoogLeNet, AlexNet, etc. to the specific domain of breast cancer detection [23]. This leads to a state where the training data, which is from the related and the specific domain, and the test data, which only comes from the specific domain with not much data, are from a different distribution [38].
Finetuning is a special case of transfer learning where the first bigger data set is obtained by oversampling the original data set. The network is then at first trained on the oversampled data set and afterwards retrained on the original data set. The reasoning is the same as for every other transfer learning task, that the bigger data set, in this case with an increased number of samples for less frequent classes, allows the network and especially the convolutions within the network to learn a better feature extraction. The finetuning in the second step should allow the network to fit to the original data distribution and in the special case of finetuning, also should reduce or eliminate the overfitting that is caused by the oversampling.
The main difference between finetuning an classical transfer learning is the size of the second data set in relation to the first data set, as well as the number of parameters in the network. For finetuning the first data set is also of course bigger than the second data set but the factor by which it is bigger, is smaller than for most transfer learning applications. This is why the number of epochs which the network is trained on the second data set as well as the learning rate that is used for the retraining need to be smaller in order to not eliminate every effect that was achieved in the first training on the oversampled data set. Additionally only the parameters in the fully connected output layer are changed during the retraining because the fully connected layer are most prone to overfitting and the feature extraction from the convolutions should not be changed.

4 Data, Methods and Evaluation

This chapter gives an overview on the specific configurations and choices that were made for this study. At first the data set and preprocessing that is applied to the data is described. Then the network architecture and other unvarying training parameters are listed. Afterwards the concrete application of the oversampling and finetuning is specified along with the parameters that are tested for these methods. In the last subsection statistical evaluation methods are assessed and the metrics chosen for evaluation in this study are presented.

4.1 Dataset

The data set consists of concentrations of nitrogen oxide (NO), nitrogen dioxide (NO_2) and ozone (O_3) as well as the cloud cover, planetary boundary layer height, relative humidity, temperature and the winds u and v component. The air quality measurements that include the various concentrations of air pollutants are given as daily moving average above an 8 hour time window and are provided by the German Umweltbundesamt. The meteorological data comes from the COSMO-REA6 [6] and for planetary boundary layer height and temperature the maximum daily value is chosen while for the other variables the average daily value is used. The data is accessible via the TOAR database [43] and a list of all stations can be found in the appendix A.
After collecting the data from all stations the data is split into training, validation and test data. The training data is then standardized to approximately mean zero and unit variance for all variables and divided into batches of 512 samples. The splitting into training, validation and test data set has to be done before the standardization to not give the network information about the validation and test data. This data preprocessing is identical to the preprocessing done by Kleinert et al. in the IntelliO3-ts paper [26]. Kleinert et al. also list a more detailed description of the data as well as the standardization process. The only differences lies in the number of days which are forecasted. While in the IntelliO3-ts paper 4 days were forecasted, this study will focus on only forecasting 1 day. The reason for this is the application of the oversampling which will be explained later in section 4.3.

4.2 Base Model and training

The network architecture used in this study is identical to the architecture used in the IntelliO3-ts paper [26]. The only small change is due to the decrease in the forecasted time steps. It consists of two inception blocks followed by two fully connected layers. For faster convergence of the learning process a minor output tail is added to the first inception block. A detailed visualization of the network architecture can be found in the appendix B.
The model without oversampling or finetuning will be referred to as “reference“ model. This reference model as well as the model where oversampling is applied will be trained for 150 epochs with a learning rate of 10^{-5} and for the backward propagation the adam optimizer [24] will be used. For the finetuning experiments, learning rate and number of

epochs are parameters that are subject to testing.
The implementation of the network, the training and the evaluation are done with MLAir [30], a machine learning framework for air time series data developed by Leufen et al., which is mainly based on keras and tensorflow.

4.3 Oversampling and Finetuning

The oversampling is applied to the training data before the training of the network takes place and it is done as follows. The range of all ozone values from the training data is divided in b identical spaced intervals called “bins“. For every bin b_i the oversampling rate o_i is determined as the number of samples in the bin, divided by the number of samples in the most frequent bin b_{max}, so $o_i = \frac{|b_i|}{|b_{max}|}$. Additionally there is a maximum oversampling rate o_{max} as an upper limit for the oversampling rates. So if any o_i would exceed this maximum it is set to o_{max} instead. Therefore the oversampling rate o_i for the bin b_i is defined as:

$$o_i = \min\left(\frac{\mid b_i \mid}{\mid b_{max} \mid}, o_{max}\right) \tag{9}$$

After the calculation of the oversampling rates, the samples in each bin are copied until the oversampling rate of the bin has been achieved. For the oversampling rate o_i, the samples should be copied $o_i - 1$ times. To achieve this, the whole sample set in the bin is copied $\lfloor o_i - 1 \rfloor$ times and the remaining $o_i - \lfloor o_i \rfloor$ rate is chosen randomly from all samples in that bin. This results in a new sample distribution where every bin has the same frequency (except for the bins where the oversampling rate is capped by o_{max}). This oversampling method has two parameters which are tested in this study, the number of bins b and the maximum oversampling rate o_{max}.
Finetuning is an additional step that is done after the oversampling and training on the oversampled training data. During finetuning the network is trained again but with the original dataset that was not oversampled. In this second training phase the learning rate of the network is lower, less epochs take place and not all parameters of the network are trainable. Instead only the parameters in the output layer are set as trainable which means only these parameters will be adjusted during backward propagation. The exact learning rate and number of epochs are parameters that are tested in this study.

4.4 Statistical Evaluation Methods

There are various evaluation metrics across climatology [16], deep learning [20] and regression analysis[7]. These fields have many overlaps, especially on the most commonly used metrics, but also some metrics that are more exclusively used in one of the research fields. This chapter first gives an overview on the most commonly used metrics and also addresses the problem they might bring for evaluating the ability to precisely forecast high or “extreme“ values. Afterwards the metrics that are used for evaluation in this study will be presented.

4.4.1 Common Evaluation Metrics

According to Botchkarev et al. [7] the most frequently used evaluation metrics for regression tasks are the mean absolute error (MAE), the mean absolute percentage error (MAPE) and the mean squared error (MSE). Let $\hat{y}$ be the prediction, y the correct label and n the number of samples:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} \mid \hat{y} - y \mid \tag{10}$$

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{\hat{y} - y}{y} \right| \tag{11}$$

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{y} - y)^2 \tag{12}$$

These metrics are all based on summation of the distances between forecast and correct value and are frequently used in climatology and especially deep learning, where the loss function which the algorithm learns from is most often one of the above mentioned metrics. The main problem with this type of metrics, especially for this study, is that the error of all samples is averaged. The MSE for example weights stronger deviation more than the MAE but it is still a calculation of the average distance, just with a slightly different distance metric. But since this study focuses on extreme ozone values which occur rarely, these types of average error functions are not well suited as evaluation metrics.

4.4.2 Metrics used in this study

The evaluation metrics and scores used in this study are mainly inspired from climatological research and are based on the very detailed overview from Wilks et al. [16]. Additionally the National Oceanic and Atmospheric Administration gives a condensed overview on forecast verification metrics [37]. The main focus in this study lies on contingency tables and multiple scores which are calculated from these. For graphical illustrations conditional quantile plots can be very useful. The main goal for the metrics is to evaluate whether the model has the ability to distinct between a high or a lower ozone concentration.

		Observations	
		y_1	y_2
Forecast	$\hat{y}_1$	a	b
	$\hat{y}_2$	c	d

Figure 5: 2×2 contingency table, with the class 1 which represents values over a certain threshold and class 2 that represents values under the threshold. a, b, c and d are the number of samples that fall under the associated category.

Contingency tables are a way to display the joint distributions of forecast and observations. They are class based, which means the forecasts and distributions are divided into two or more classes. For this study 2 × 2 contingency tables will be build by setting a threshold and dividing observations and forecasts in two groups, group 1 is above the threshold and and group 2 below. This means that y_1 are the observations above the threshold, y_2 the observations below the threshold and the same goes for the forecasts for $\hat{y_1}$ and $\hat{y_2}$ (fig. 5). There are many scores that can be calculated based on contingency tables. The scores in general compress the information of the contingency table to a single value. Obtaining single values from contingency tables makes it possible to plot them for every threshold, resulting in a plot of the score in dependence of the threshold.

The gilbert skill score (GSS) is well suited for the overall aim of this study. It uses the non random correct forecasts above the threshold, which means the correct forecasts above the threshold a, minus the hits by chance ch, calculated as the frequency of the event times the number of forecasts for this event. This is divided by the sum of correct forecasts above the threshold a, false forecasts above the threshold b and false forecasts below the threshold c minus the hits by chance ch:

$$GSS = \frac{a - ch}{a + b + c - ch} \tag{13}$$

This gives a metric to evaluate the networks ability to reliably forecast whether a certain threshold is exceeded or not, while also accounting for random guesses and also the false forecasts. The gilbert skill score additionally has the advantage, that it works well for less frequent classes since the correct forecasts below the threshold are not taken into account. Additionally the hit rate (H), the false alarm rate (F) and the bias (B) are chosen because these three combined allow a reconstruction of a complete contingency table and therefore capture all information provided by the table.

$$H = \frac{a}{a + c} \tag{14}$$

$$F = \frac{b}{b + d} \tag{15}$$

$$B = \frac{a + b}{a + c} \tag{16}$$

Overall the gilbert skill score and hit rate are indicators to whether the aim of the study, increasing accuracy on extreme values, is achieved. The main difference between the two is that the hit rate only focuses on the percentage of correctly forecasted samples that are above the threshold, while the gilbert skill score also accounts for the samples that are falsely forecasted to be over the threshold. Ideally especially with growing thresholds, the gilbert skill score and hit rate from the models trained in this study should be higher than the scores from the reference model. False alarm rate and bias on the other hand can display the drawbacks the applied methods might have in sometimes falsely predicting extreme values. Therefore gilbert skill score and hit rate can be categorized as “positive oriented“ metrics where a higher score is good while false alarm rate and bias are “negative oriented“ metrics where a lower score is desirable. To prevent dividing by zero and to make the scores more smooth every contingency cell is increased by one, therefore avoiding null entries in the contingency cells.

Conditional quantile plots are a method to graphically illustrate the joint distribution from observations and forecasts. They consist of two parts. The first is a histogram at the bottom that displays the forecast distribution. The second part displays the conditional distribution of the observations with a given forecast. It consists of several lines that represent different quantiles as well as the 1:1 reference line which represents a perfect forecast. The reference line is important to detect systematic over- or underestimations. If the 50% quantile is below the reference line, this implies that the mean observation for this prediction value is below that value. This signals systematic overestimation since on average the observation given this predicted value is lower than the given prediction. The convergence of the different quantiles, especially at the upper end of the forecasted values, can have two reasons. Either the distribution of the observations given a certain threshold just has low variance or as it can be observed in the example in figure 6 the number of forecasts for this concentration is so low, that the borders of the different quantiles merge and one sample represents multiple quantiles. The histogram that displays the forecast distribution can overall be seen as a indicator for the statistical significance of the conditional quantiles. If the histogram shows many forecasts for a single value, the conditional quantiles for that value are more reliable.

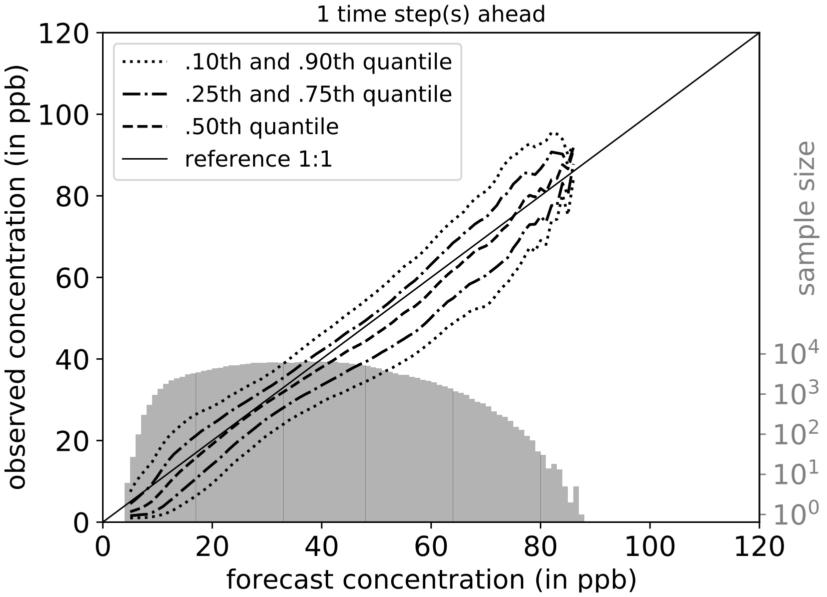


Figure 6: Example of a conditional quantiles plot with the observed concentrations in dependence on the forecasted concentrations. On the bottom a histogram of the forecast distribution is displayed with a logarithmic scale (on the right axis). From bottom to top right the reference line as well as the different quantiles of observations for a given forecast value are illustrated.

5 Results

In this chapter the conducted experiments will be listed. They can be divided into two parts, experiments with oversampling (sect. 5.1) and experiments with oversampling and subsequent finetuning (sect. 5.2). In both parts a hyperparameter search is done for the method specific parameters which means that multiple parameter values are tested and evaluated against each other. The evaluation and comparison of the different parameter values will be mainly based on the contingency table score plots. Additionally for the thresholds of 60 ppb and 80 ppb exact values for the different scores and models will be used as a case study. To assess overall performance and to showcase the influence on the joint distribution of forecasts and observations, conditional quantile plots will be displayed for specific experiments.

The experiments listed in this section do not contain every model and every parameter value that was tested in the study but only the models with reasonable results and significant changes are displayed. Furthermore, some of the oversampling experiments as well as the finetuning experiments were conducted multiple times with the exact same parameters to investigate the volatility in the results and test their statistical significance. The deviations in the contingency scores between models with identical parameters where always less than 0.5%.

5.1 Oversampling

The two parameters that can be adjusted for the oversampling are the maximum oversampling rate and the number of bins that the range of ozone concentrations is split into (see sect. 4.3). First, different oversampling rates will be tested while using a fixed number of bins (sect. 5.1.1) and then vice versa the number of bins will be changed while using a constant maximum oversampling rate (sect. 5.1.2). In the end an experiment with different oversampling rates as well as different number of bins will be conducted (sect. 5.1.3). For all experiments the network that was trained without oversampling will be used for reference (see sect. 4.2).

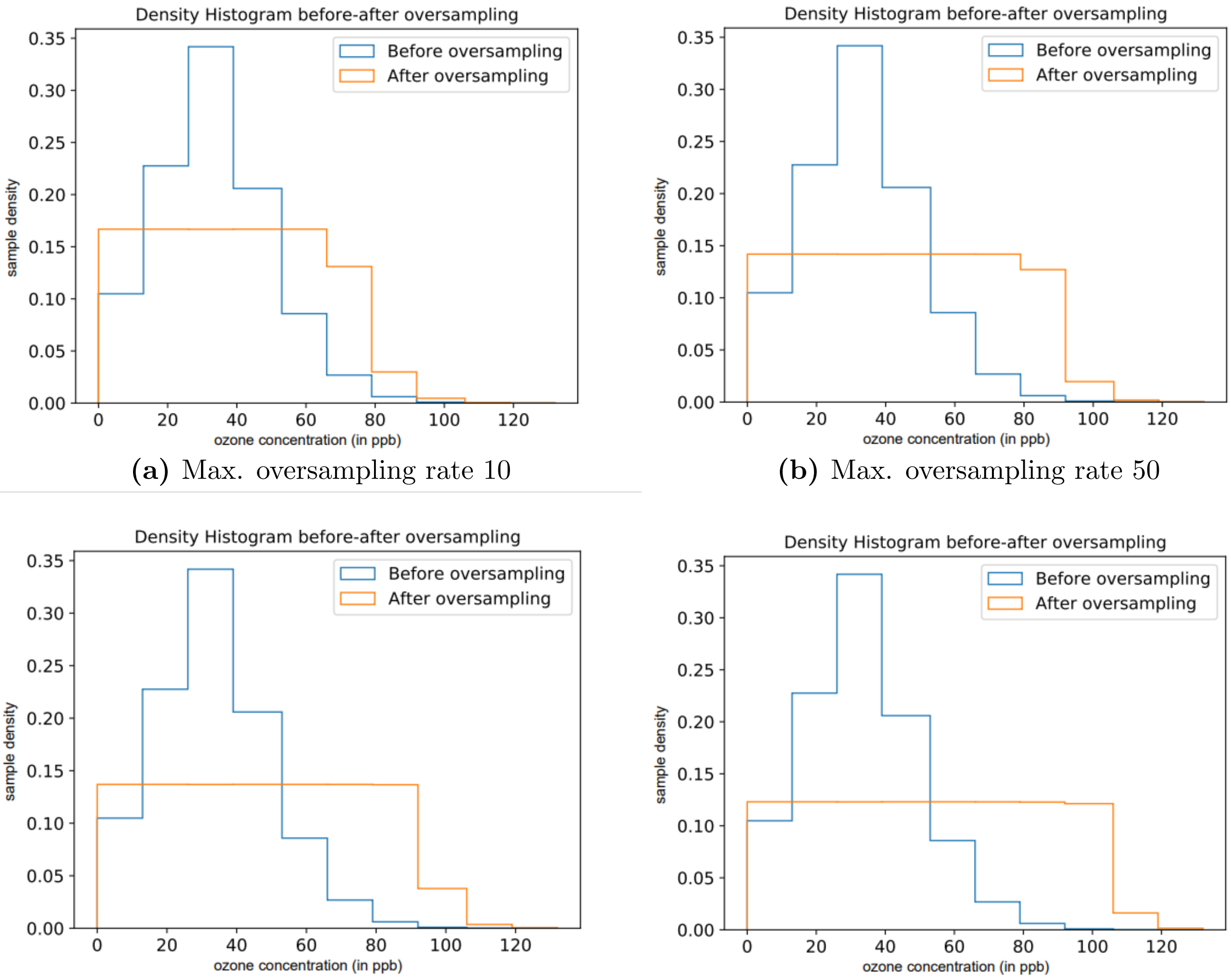
5.1.1 Different oversampling rates

For this experiment the number of bins will be fixed to 10. The maximum oversampling rates used are 10, 50, 100 and 500. The different maximum oversampling rates lead to different success in the flattening of the training sample distribution. Table 1 displays the number of samples that are in each of the 10 bins as well as the frequency and the oversampling rate that would be needed to fully balance the training data.

Table 1: Number of samples, frequency and oversampling rate that would be necessary to completly balance the data for each of the 10 bins that the training data is split into.

Bin	1	2	3	4	5	6	7	8	9	10
Samp.	77946	169155	254073	153046	63795	19966	4561	712	77	16
Freq.	0.1049	0.2276	0.3418	0.2059	0.0858	0.0269	0.0061	0.00096	0.00010	0.00002
Rate	3.2496	1.5020	1.0000	1.6601	3.9826	12.725	55.706	356.84	3299.6	15880

Figure 7 shows density histograms for all four maximum oversampling rates before and after the oversampling. The higher the maximum oversampling rate, the more bins reach the same frequency. For a maximum oversampling rate of 10 only 5 out of 10 bins have the same frequency (fig. 7a) while for the maximum oversampling rate of 500 (almost) 8 bins contain the same number of samples (fig. 7d). But even for the highest oversampling rate the frequency of the last bin is rarely visible since the oversampling rate that would be needed for this bin to contain the same amount of samples would be 15880 (tab. 1).



(a) Max. oversampling rate 10

(b) Max. oversampling rate 50

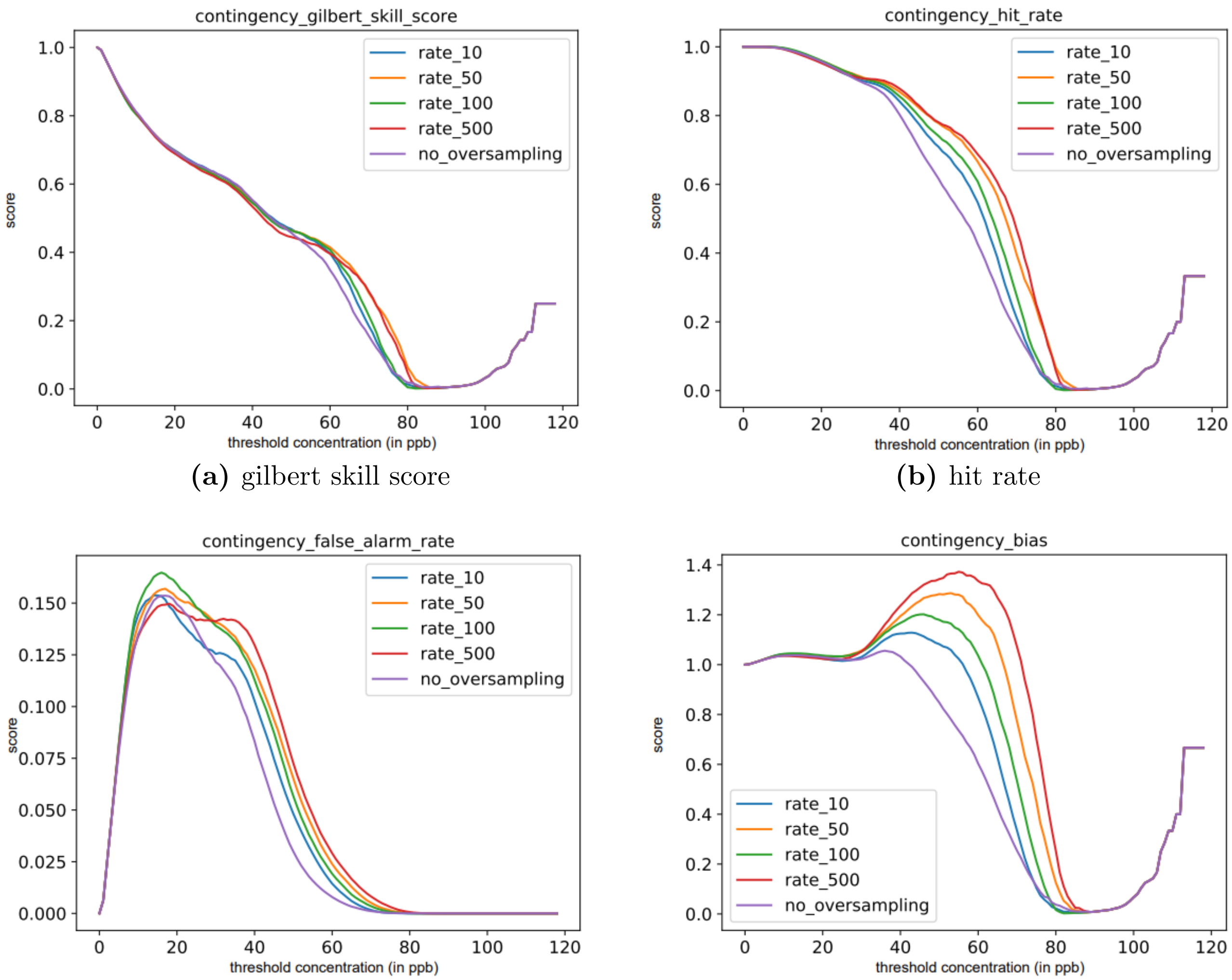
(c) Max. oversampling rate 100

(d) Max. oversampling rate 500

Figure 7: Density histograms for all four oversampled models with different max. oversampling rates. The intervals of the histograms are equal to the intervals that are used for the bins.

On each of the oversampled data sets that are displayed in figure 7, a model was trained and figure 8 shows the plots of the different contingency scores of each model. All four plots have two things in common. First is that the model without oversampling has the lowest scores for all thresholds, with only one exception at the false alarm rate between

15 ppb and 30 ppb (fig. 8c). Second is that there is always only one interval of thresholds where the oversampled models deviate from the model without oversampling. For the gilbert skill score this interval is between 50 ppb and 85 ppb (fig. 8a), hit rate and bias differ for thresholds between 30 ppb and 85 ppb (fig. 8b, 8d) and for the false alarm rate this window is between 15 ppb and 75 ppb (fig. 8c). The models with oversampling rates of 50 and 500 show the highest scores for almost all thresholds. For the positive oriented metrics they are closely together while the false alarm rate and bias are noticeably higher for the model with the maximum oversampling rate of 500.



(a) gilbert skill score

(b) hit rate

(c) false alarm rate

(d) bias

Figure 8: Threat score, hit rate, false alarm rate and bias in dependence of the ozone concentration threshold in ppb. The different models were trained with oversampling, 10 bins and a maximum oversampling rate of 10, 50, 100 and 500. A model without oversampling is used for reference.

This effect can also be seen in tables 2 and 3 where the scores for the different models are listed at the thresholds of 60 ppb and 80 ppb. They again highlight how close the models with maximum oversampling rate of 50 and 500 are. For 60 ppb, all oversampled models are fairly close together while for 80 ppb, there are bigger discrepancies between the models. The model with a maximum oversampling rate of 50 has the highest gilbert skill score for both thresholds and also the highest hit rate at a threshold of 80 ppb. The model with an oversampling rate of 500 has the highest hit rate for a threshold of 60 ppb as well as the highest false alarm rate and bias for both thresholds.

Table 2: The different scores: gilbert skill score (GSS), hit rate (H), false alarm rate (F) and bias (B) for the different models with maximum oversampling rate of 10, 50, 100 and 500 and 10 bins and the model without oversampling at a threshold of 60 ppb. The highest value for each score is marked in bold face. At the bottom the absolute and relative increase from the model with the highest score to the model without oversampling are listed.

Threshold: 60 ppb	GSS	H	F	B
rate 10	0.3976	0.5476	0.0144	0.8685
rate 50	**0.4150**	0.6661	0.0239	1.1987
rate 100	0.4078	0.6085	0.0192	1.0361
rate 500	0.3961	**0.6874**	**0.0291**	**1.3357**
no oversampling	0.3482	0.4266	0.0079	0.6031
absolute increase	0.0668	0.2608	0.0212	0.7326
relative increase	0.1917	0.6114	2.6742	1.2147

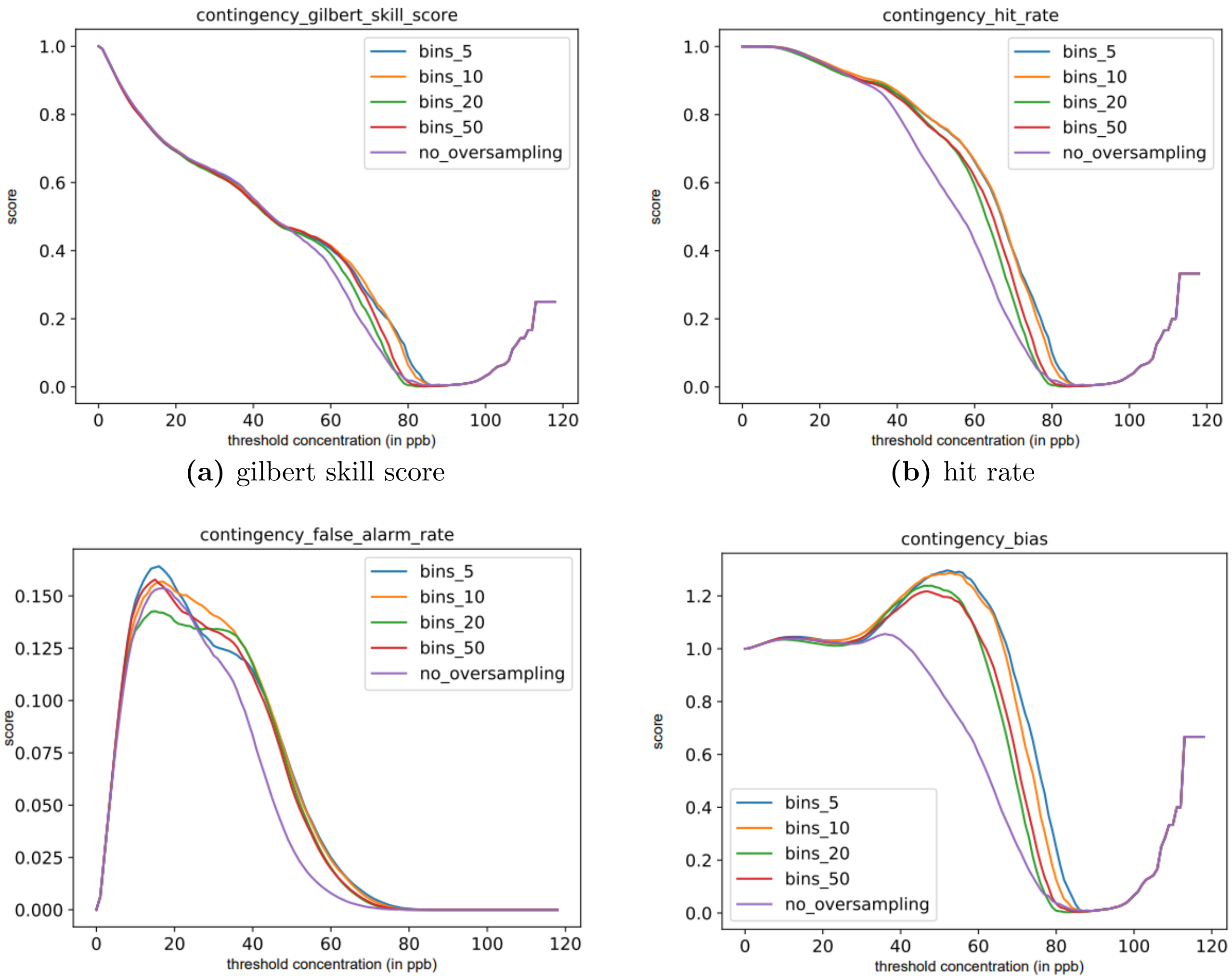
Table 3: The different scores: gilbert skill score (GSS), hit rate (H), false alarm rate (F) and bias (B) for the different models with maximum oversampling rate of 10, 50, 100 and 500 and 10 bins and the model without oversampling at a threshold of 80 ppb. The highest value for each score is marked in bold face. At the bottom the absolute and relative increase from the model with the highest score to the model without oversampling are listed.

Threshold: 80 ppb	GSS	H	F	B
rate 10	0.0131	0.0133	0.00003	0.0218
rate 50	**0.0634**	**0.0674**	0.00020	0.12536
rate 100	0.0046	0.0048	0.00004	0.0161
rate 500	0.0478	0.0579	**0.00068**	**0.2526**
no oversampling	0.0185	0.0190	0.00006	0.0370
absolute increase	0.0448	0.0484	0.00062	0.2156
relative increase	2.4188	2.5508	10.3333	5.8199

At a threshold of 60 ppb the scores imply that the model with a maximum oversampling rate of 500 would correctly classify 69% of the samples that are above the threshold while the model without oversampling only achieves this for 43%. On the other hand the oversampled model would falsely predict 2.9% of the samples below the threshold as above the threshold while this only happens for 0.8% of the samples with the model without oversampling. In absolute terms this means an increase of 0.26 in the hit rate at the cost of increasing the false alarm rate by 0.021 while in relative terms the hit rate gets increased by 61% while the false alarm rate gets increased by 267%. So the absolute increase is higher for the hit but the relative increase is higher for the false alarm rate. This is also true for the 80 ppb threshold where the relative increase is overall higher than for 60 ppb but the absolute increase is lower.

5.1.2 Different number of bins

For this experiment the maximum oversampling rate is fixed to 50 while the number of bins are 5, 10, 20 and 50.



(a) gilbert skill score

(b) hit rate

(c) false alarm rate

(d) bias

Figure 9: Gilbert skill score, hit rate, false alarm rate and bias in dependence of the ozone concentration threshold in ppb. The different models were trained with oversampling, a maximum oversampling rate of 50 and 5, 10, 20 or 50 bins. A model without oversampling is used for reference.

The score plots in figure 9 show similar characteristics to the plots in the previous experiment. The model without oversampling again has the lowest scores for all thresholds, except for the false alarm rate between 10 ppb and 30 ppb (fig. 9c). Deviations between the models are also again only in a certain range of thresholds and these ranges are also identical to the previous experiment, from 50 ppb to 85 ppb for the gilbert skill score (fig. 9a), from 30 ppb to 85 ppb for hit rate and bias (fig. 9b, 9d) and from 15 ppb to 75 ppb for the false alarm rate (fig. 9c). The models with 5 or 10 bins have the highest score for almost all thresholds, both for the positive as well as the negative oriented metrics.

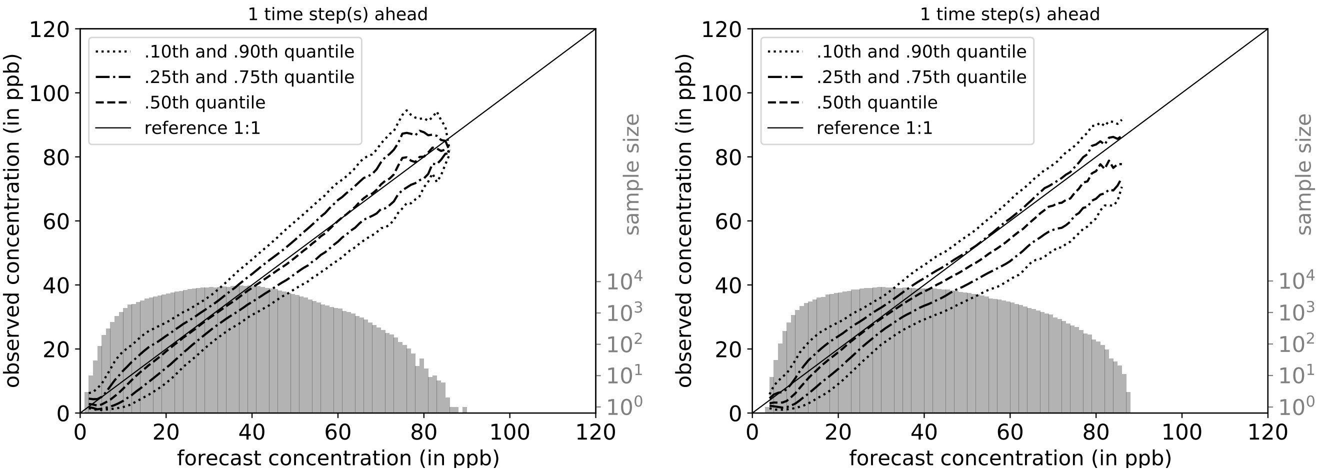
Table 4: The different scores: gilbert skill score (GSS), hit rate (H), false alarm rate (F) and bias (B) for the different models with 5, 10, 20 and 50 bins and a maximum oversampling rate of 50 and the model without oversampling at a threshold of 60 ppb. The highest value for each score is marked in bold face. At the bottom the absolute and relative increase from the model with the highest score to the model without oversampling are listed.

Threshold: 60 ppb	GSS	H	F	B
bins 5	0.4052	0.6620	**0.0250**	**1.2188**
bins 10	**0.4150**	**0.6661**	0.0239	1.1987
bins 20	0.3903	0.5928	0.0202	1.0419
bins 50	0.4110	0.6173	0.0196	1.0541
no oversampling	0.3482	0.4266	0.0079	0.6031
absolute increase	0.0668	0.2395	0.0171	0.6157
relative increase	0.1917	0.5615	2.1553	1.0209

Table 5: The different scores: gilbert skill score (GSS), hit rate (H), false alarm rate (F) and bias (B) for the different models with 5, 10, 20 and 50 bins and a maximum oversampling rate of 50 and the model without oversampling at a threshold of 80 ppb. The highest value for each score is marked in bold face. At the bottom the absolute and relative increase from the model with the highest score to the model without oversampling are listed.

Threshold: 80 ppb	GSS	H	F	B
bins 5	**0.0871**	**0.1007**	**0.00051**	**0.2469**
bins 10	0.0634	0.0674	0.00020	0.1254
bins 20	0.0047	0.0048	0.00001	0.0086
bins 50	0.0148	0.0152	0.00007	0.0351
no oversampling	0.0186	0.0190	0.00006	0.0370
absolute increase	0.0686	0.0817	0.00045	0.2099
relative increase	3.7026	4.3012	7.5000	5.6660

Tables 4 and 5 also underline the observations from the score plots. For the 60 ppb threshold the scores from the oversampled model are very close together while for the 80 ppb threshold, the differences are more clearly. At the 60 ppb threshold the model with 10 bins has the (slightly) highest gilbert skill score and hit rate and the model with 5 bins has the highest false alarm rate and bias. The latter has the highest scores in all four metrics at a threshold of 80 ppb. Just like in the last experiment (tab. 2, 3) the hit rate has a higher increase in absolute terms while the false alarm rates increase is relatively higher. For 60 ppb the model with an oversampling rate of 50 and 10 bins can capture 67% of the threshold exceedings while the model without oversampling only correctly forecasts 43% of them. In return the false alarm rate is also increased from 0.8% to 2.4%. Figure 10 displays the conditional quantile plots for the experiment without oversampling (fig. 10a), and the experiment with a maximum oversampling rate of 50 and 5 bins (fig. 10b). The overall forecast distribution of the model with oversampling shows an increase in forecasts of higher values, especially in the range from 80 ppb to 85 ppb and an overall more equal distribution. While the 50% quantile line for the model without oversampling is very close to the reference line, the 50% quantile line for the model with oversampling drops below the reference line at 40 ppb and stays below it for all forecasts above 40 ppb. This signals that the model with oversampling on average overestimates ozone concentrations when it forecasts above 40 ppb. The difference at the upper end of the quantile lines, where they converge only for the model without oversampling, can be explained by the forecast distributions. The model without oversampling only forecasts the highest values once, while the model with oversampling does it multiple times, which leads to not converging quantiles as explained in section 4.4.2.



(a) without oversampling

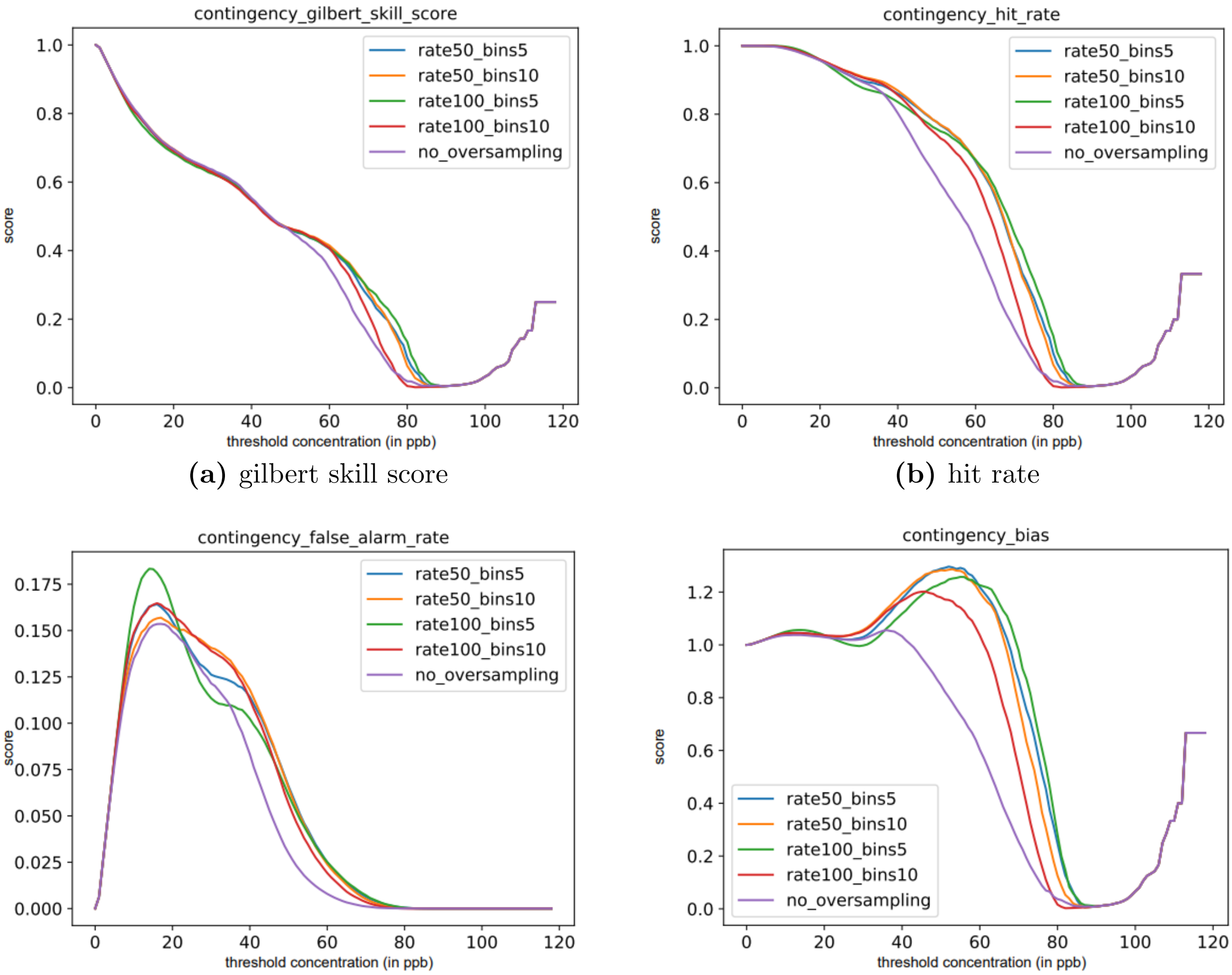
(b) oversampling rate 50, 5 bins

Figure 10: Conditional quantile plots for the model without oversampling (on the left) and the model a maximum oversampling rate of 50 and 5 bins (on the right). At the bottom the forecast distribution is displayed in a logarithmic scale and from bottom left to top right the observation distribution and its quantiles given a certain forecast value are shown.

5.1.3 Different oversampling rates and number of bins

In this final experiment in the oversampling section, four different combinations of maximum oversampling rates of 50 and 100, paired with 5 and 10 bins will be compared.

These combinations were chosen because these parameters showed the highest or one of the highest scores in the two previous experiments (sect. 5.1.1, 5.1.2).



(a) gilbert skill score

(b) hit rate

(c) false alarm rate

(d) bias

Figure 11: Gilbert skill score, hit rate, false alarm rate and bias in dependence of the ozone concentration threshold in ppb. The four models were trained with oversampling, with the possible combinations of maximum oversampling rate of 50 and 100 and 5 or 10 bins. A model without oversampling is used for reference.

The characteristics of the score plots in figure 11 are again very similar to the previous experiments both regarding the model without oversampling having the lowest scores and the threshold ranges in which the scores are different. The gilbert skill score (fig. 11a) and the hit rate (fig. 11b) are highest for the model with maximum oversampling rate of 100 and 5 bins, especially for higher thresholds. This model also has the highest false alarm rate (fig. 11c) for thresholds in the range of 10 ppb to 25 ppb and again from 45 ppb to 80 ppb while between these two ranges it is also slightly lower than the model without oversampling. Lastly the model with maximum oversampling rate of 100 and 5 bins also has the highest bias (fig. 11d) in the range of 65 to 90 while prior to this range the models with maximum oversampling rate of 50 and 5 or 10 bins have the highest bias.

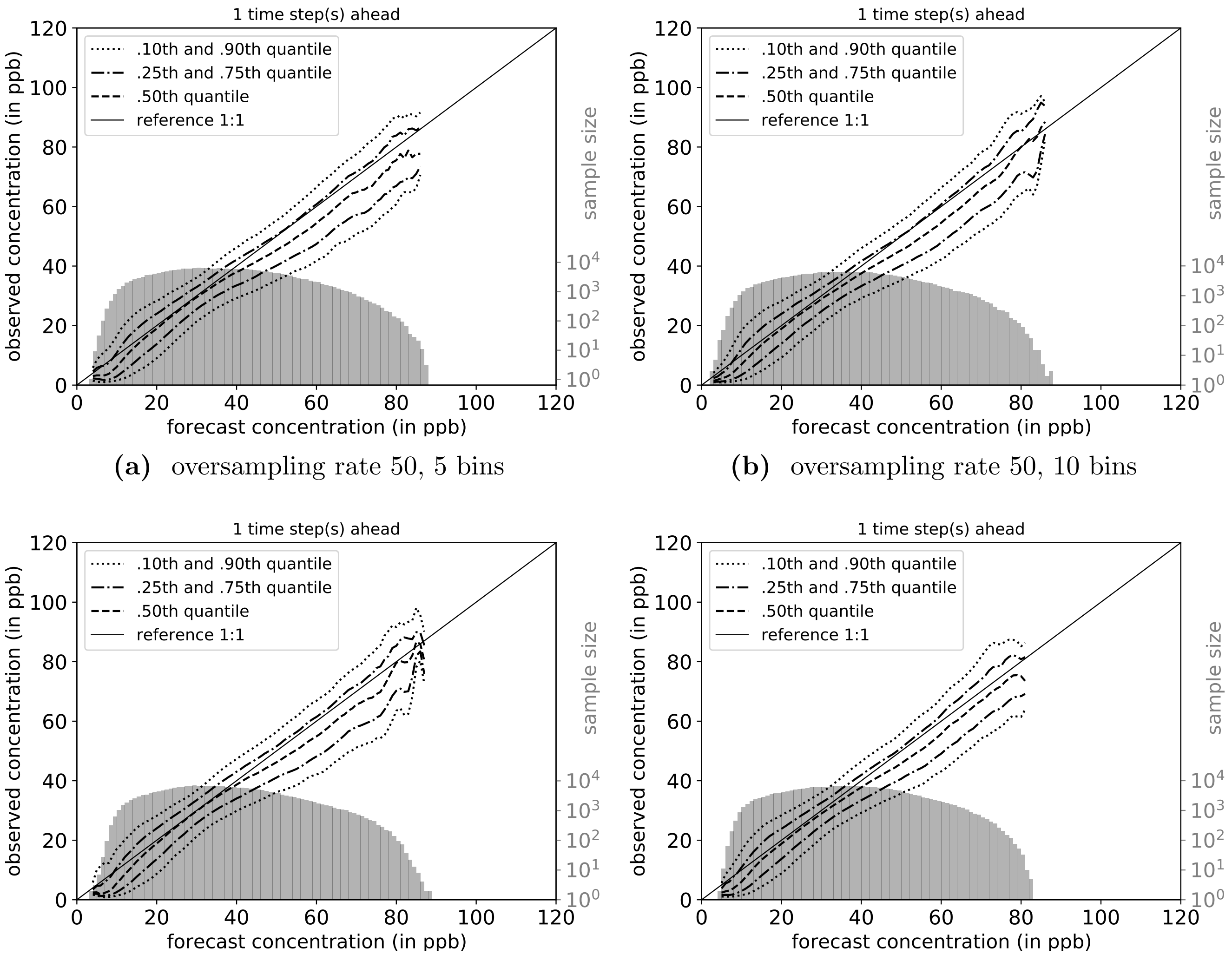
Table 6: The different scores: gilbert skill score (GSS), hit rate (H), false alarm rate (F) and bias (B) for the different models with maximum oversampling rates of 50 and 100 and 5 or 10 bins and the model without oversampling at a threshold of 60 ppb. The highest value for each score is marked in bold face. At the bottom the absolute and relative increase from the model with the highest score to the model without oversampling are listed.

Threshold: 60 ppb	GSS	H	F	B
rate 50, bins 5	0.4052	0.6620	0.0250	1.2188
rate 50, bins 10	**0.4150**	0.6661	0.0239	1.1987
rate 100, bins 5	0.4074	**0.6665**	**0.0251**	**1.2260**
rate 100, bins 10	0.4078	0.6085	0.0192	1.0361
no oversampling	0.3482	0.4266	0.0079	0.6031
absolute increase	0.0668	0.2399	0.0172	0.6229
relative increase	0.1917	0.5624	2.1705	1.0328

Table 7: The different scores: gilbert skill score (GSS), hit rate (H), false alarm rate (F) and bias (B) for the different models with maximum oversampling rates of 50 and 100 and 5 or 10 bins and the model without oversampling at a threshold of 80 ppb. The highest value for each score is marked in bold face. At the bottom the absolute and relative increase from the model with the highest score to the model without oversampling are listed.

Threshold: 80 ppb	GSS	H	F	B
rate 50, bins 5	0.0871	0.1007	**0.00051**	0.2469
rate 50, bins 10	0.0634	0.0674	0.00020	0.1254
rate 100, bins 5	**0.1338**	**0.1520**	0.00045	**0.2811**
rate 100, bins 10	0.0046	0.0048	0.00004	0.0161
no oversampling	0.0185	0.0190	0.00006	0.0370
absolute increase	0.1153	0.1330	0.00045	0.2441
relative increase	6.2218	7.0016	7.5000	6.5891

The tables 6 and 7 display the score at the 60 ppb and 80 ppb threshold. They show that the model with a maximum oversampling rate of 100 and 5 bins has the highest score for almost all metrics. Only exceptions are the highest gilbert skill score at 60 ppb which is obtained by the model with maximum oversampling rate 50 and 10 bins and the highest false alarm rate at 80 ppb from the model with maximum oversampling rate 50 and 5 bins. The model with oversampling rate 100 and 5 bins achieves an absolute increase in the hit rate of 0.24 at 60 ppb and 0.13 at 80 ppb while only increasing the false alarm rate for 0.017 and 0.00045 respectively. Especially the increase at 80 ppb which is 700% is significantly higher than for any other model. This means that the model with an oversampling rate of 100 and 5 bins has the ability to correctly forecasted 67% instead of only 43% of the samples above 60 ppb and 15.1% instead of 1.9% of the samples above 80 ppb. Just like in the previous two experiments (tab. 2, 3, 4, 5), the scores from the oversampled models are very close together at the threshold of 60 ppb while having bigger discrepancies for the threshold of 80 ppb.



(a) oversampling rate 50, 5 bins

(b) oversampling rate 50, 10 bins

(c) oversampling rate 100, 5 bins

(d) oversampling rate 100, 10 bins

Figure 12: Condtional quantile plots for the models with maximum oversampling rate of 50 and 5 bins, rate of 50 and 10 bins, rate of 100 and 5 bins and rate of 100 and 10 bins (bottom right). At the bottom the forecast distribution is displayed in a logarithmic scale and from bottom left to top right the observation distribution and its quantiles given a certain forecast value are shown.

Figure 12 displays conditional quantile plots for all four models with oversampling that were compared in this experiment. Differences between the models can be observed especially in the upper end of the forecast distribution as well as in the systematic overestimation, indicated by the 50% quantile being below the reference line. The model with a maximum oversampling rate of 50 and 5 bins (fig. 12a) has the most forecasts in the range from 80 ppb to 85 ppb while the forecast distribution of the model with a maximum oversampling rate of 100 and 10 bins (fig. 12d) ends earliest. In terms of systematic overestimation, the 50% quantile of these two models is below the reference line for all concentrations above 30 ppb while for the other two models (fig. 12b, 12c) the 50% quantile converges back to the reference line at the end of the forecast distribution. This indicates that all models on average overestimate forecasts above 30 ppb but for the models with oversampling rate 50 and 10 bins, and rate 100 and 5 bins, this overestimation vanishes at the upper end of the forecast distribution.

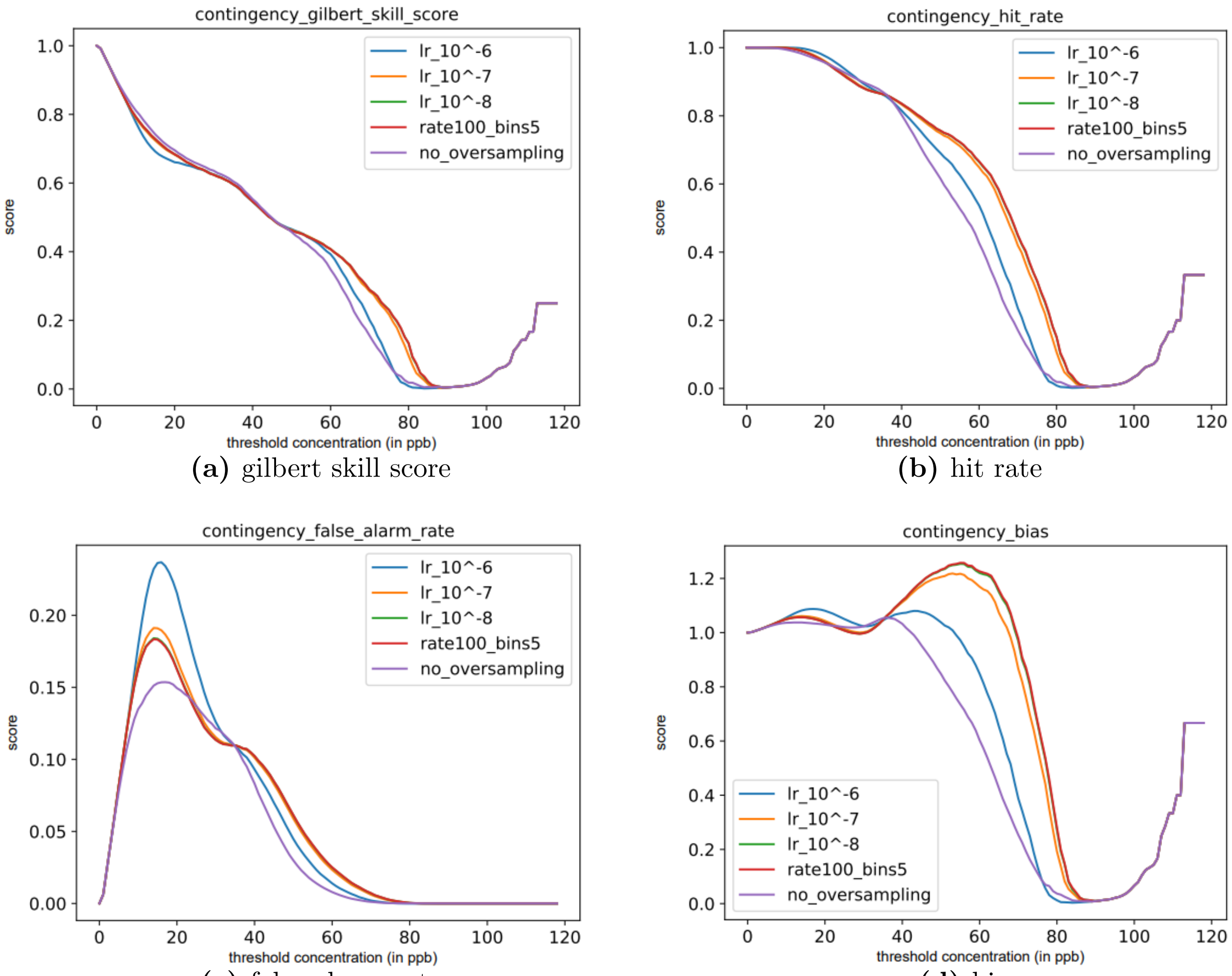
In summary, the oversampling shows an impact on the contingency scores, increasing both positive and negative oriented metrics (fig. 8, 9, 11). The conditional quantile plots (fig. 10, 12) also show a difference for the oversampled models, especially with an increase in systematic overestimation and an increase in higher forecasts, leading to a more equal forecast distribution. Concerning specific oversampled models, the model with a maximum oversampling rate of 100 and 5 bins shows the overall best scores in the positive oriented metrics while not having substantially higher scores for the negative oriented metrics. Especially for the threshold of 80 ppb it shows significant advantages over the other oversampled models (tab. 7). This is why this model will be used for the finetuning in the following section.

5.2 Finetuning

In this section the model that was trained with a maximum oversampling rate of 100 and 5 bins will be taken as a base model and finetuned according to different parameters. Finetuning is a retraining of the oversampled model using the original data set without oversampling but only the parameters in the main output layer are retrained (see sect. 4.3). In the first experiment different learning rates will be used with a fixed number of epochs (sect. 5.2.1) and in the second experiment the learning rate will be fixed and the training is done for different number of epochs (sect. 5.2.2). For the score plots the base model as well as the model without oversampling will be used for reference.

5.2.1 Different learning rates

In this experiment the base model will be finetuned for 10 epochs with learning rates of 10^{-6}, 10^{-7} and 10^{-8} (see eq. 4). Figure 13 shows the contingency score plots for the different models. In contrast to the experiments in the previous section the order of the score values below 40 ppb is very different to the order above 40 ppb. Above 40 ppb all scores show a consistent result with the model without oversampling having the lowest scores and the base model having the highest score. The higher the learning rate of the finetuned models, the lower their scores are and therefore the closer they are to the model without oversampling. The model with a learning rate of 10^{-8} has such a low learning rate that it shows almost no change to the base model. For the threshold range between 10 ppb and 40 ppb the results are not that concordant, especially the false alarm rate (fig. 13c) shows very strong deviations and the model with a learning rate of 10^{-6} has a noticable higher false alarm rate than the other models.



(a) gilbert skill score

(b) hit rate

(c) false alarm rate

(d) bias

Figure 13: Gilbert skill score, hit rate, false alarm rate and bias in dependence of the ozone concentration threshold in ppb. The four models were first trained with oversampling and a maximum oversampling rate of 100 and 5 bins. Three of the models where then finetuned for 10 epochs with learning rates of 10^{-6}, 10^{-7} and 10^{-8}. A model that was trained without oversampling is used for reference.

Tables 8 and 9 display the scores of the different models at the thresholds of 60 ppb and 80 ppb. It shows how close the base model and the model with a learning rate of 10^{-8} are and that one of those two has the highest score for both thresholds and all scores. The model without oversampling has the lowest scores at 60 ppb while the model that was finetuned with a learning rate of 10^{-6} has the lowest scores at 80 ppb. For both thresholds and all scores, the model with a learning rate of 10^{-7} is closer to the model with a learning rate of 10^{-8} than to the model with a learning rate of 10^{-6}.

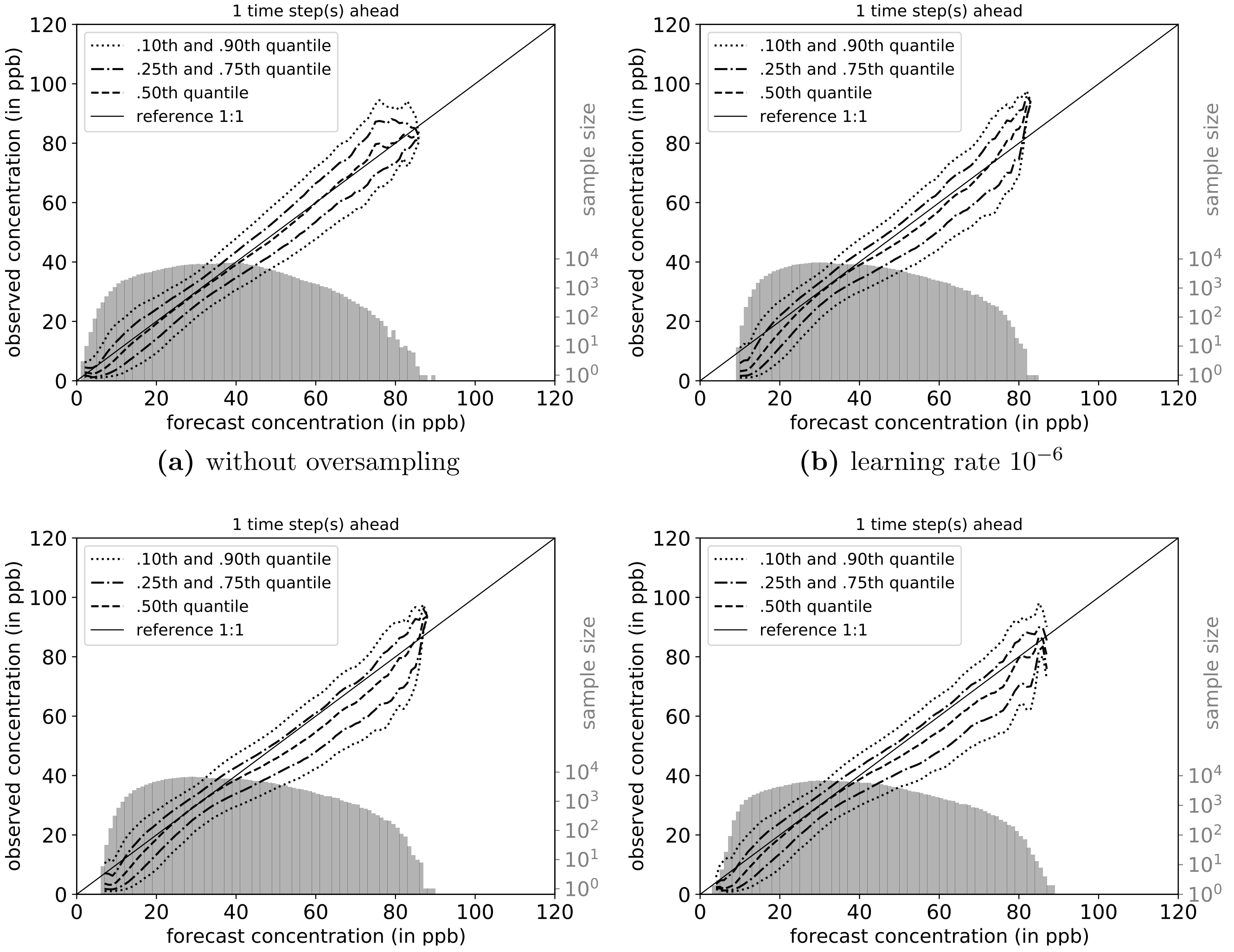
Table 8: The different scores: gilbert skill score (GSS), hit rate (H), false alarm rate (F) and bias (B) for the different models with out oversampling, with maximum oversampling 100 and 5 bins and finetuning of this model with a learning rate of 10^{-6}, 10^{-7} and 10^{-8} and 10 epochs at a threshold of 60 ppb. The highest value for each score is marked in bold face.

Threshold: 60 ppb	GSS	H	F	B
lr 10^{-6}	0.3927	0.5371	0.0140	0.8482
lr 10^{-7}	0.4074	0.6491	0.0234	1.1693
lr 10^{-8}	**0.4074**	0.6647	0.0249	1.2199
rate 100, bins 5	0.4074	**0.6665**	**0.0251**	**1.2260**
no oversampling	0.3482	0.4266	0.0079	0.6031

Table 9: The different scores: gilbert skill score (GSS), hit rate (H), false alarm rate (F) and bias (B) for the different models with out oversampling, with maximum oversampling 100 and 5 bins and finetuning of this model with a learning rate of 10^{-6}, 10^{-7} and 10^{-8} and 10 epochs at a threshold of 80 ppb. The highest value for each score is marked in bold face.

Threshold: 80 ppb	GSS	H	F	B
lr 10^{-6}	0.0085	0.0086	0.00001	0.0124
lr 10^{-7}	0.0974	0.1064	0.00030	0.1928
lr 10^{-8}	0.1331	0.1501	0.00042	0.2716
rate 100, bins 5	**0.1338**	**0.1520**	**0.00045**	**0.2811**
no oversampling	0.0185	0.0190	0.00006	0.0370

Figure 14 shows the conditional quantile plots for the model without oversampling (fig. 14a), the finetuned models with a learning rate of 10^{-6} (fig. 14b) and 10^{-7} (fig. 14c) and the base model (fig. 14d). This order also matches the order of the scores for most thresholds, with the model without oversampling having the lowest scores for most thresholds and the base model having the highest scores. This trend can also be observed in the systematic overestimation between 40 ppb and 75 ppb. While the 50% quantile of the model without oversampling is very close to the reference line in this range, it goes below that reference line for the other models and the distance between reference line and 50% quantile increases with decreasing learning rate. The distribution of the forecasts from the finetuned models (fig. 14b, 14c) is more compact, meaning that the ranges of forecasted ozone concentrations is shorter, both on the upper and especially the lower end of the distribution. This shortening of the forecast range is stronger for the model with the higher learning rate. Another distinctiveness of the finetuned models is, that above 80 ppb, all quantile lines go above the reference lines which signals that an underestimation is happening at this forecast level.



(a) without oversampling

(b) learning rate 10^{-6}

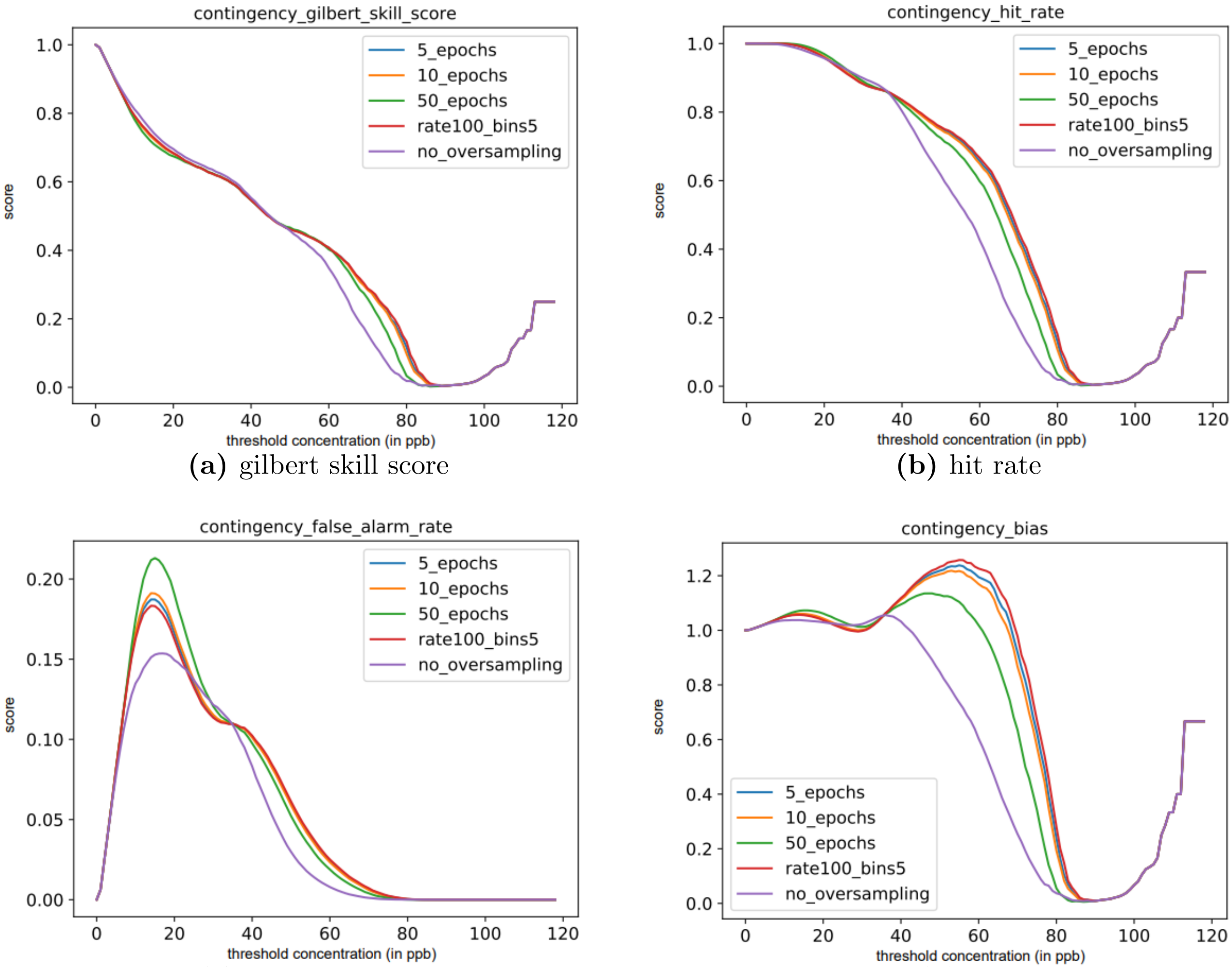
(c) learning rate 10^{-7}

(d) base model

Figure 14: Conditional quantile plots for the model without oversampling (top left), the finetuned models with a learning rate of 10^{-6} (top right) and 10^{-7} (bottom left) and the model with maximum oversampling rate 100 and 5 bins(bottom right) (bottom right). At the bottom the forecast distribution is displayed in a logarithmic scale and from bottom left to top right the observation distribution and its quantiles given a certain forecast value are shown.

5.2.2 Different Number of Epochs

In this experiment, the base model with a maximum oversampling rate of 100 and 5 bins will again be finetuned, but this time with a fixed learning rate of 10^{-7} and 5, 10 and 50 epochs. The patterns shown by the contingency score plots in figure 15 are very similar to the score plots in the previous section (fig. 13). For the range from 40 ppb to 85 ppb, the base model has the highest scores, the model without oversampling has the lowest and the higher the number of epochs, the lower the score. For the false alarm rate (fig. 15c) between 10 ppb and 40 ppb the result is also very similar to the previous experiment (fig. 13c). The finetuned models have higher scores than the reference and the base model and the higher the number of epochs, or the higher the learning rate as it was in the last experiment, the higher the false alarm rate.



(a) gilbert skill score

(b) hit rate

(c) false alarm rate

(d) bias

Figure 15: Gilbert skill score, hit rate, false alarm rate and bias in dependence of the ozone concentration threshold in ppb. The four models were first trained with oversampling and a maximum oversampling rate of 100 and 5 bins. Three of the models were then finetuned with a learning rate 10^{-7} for 5, 10 and 50 epochs. A model that was trained without oversampling is used for reference.

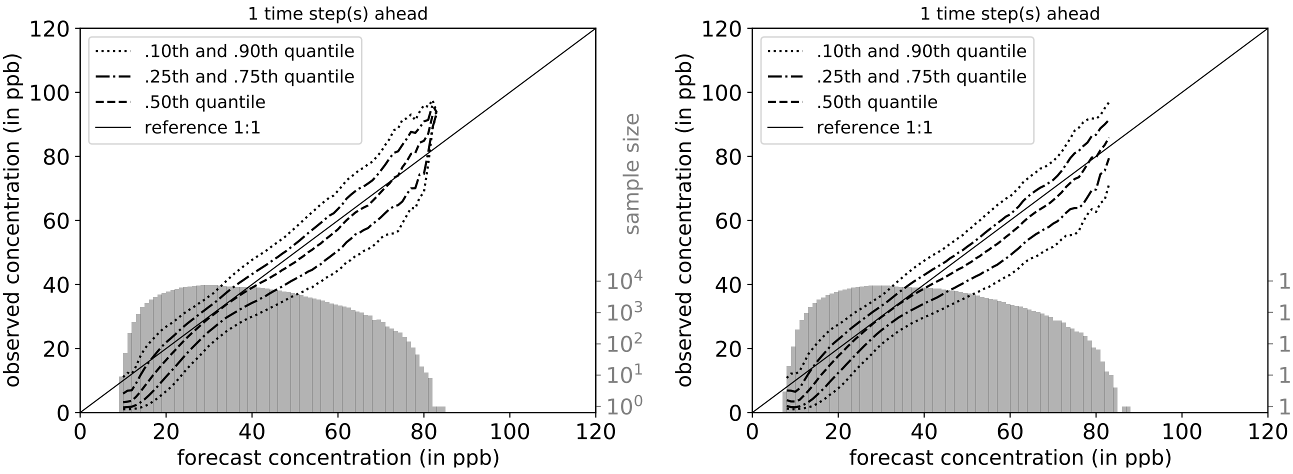
Tables 10 and 11 display the scores for the different models at 60 ppb and 80 ppb and are also in line with the characteristics of the score plots (fig. 15). The model that was finetuned for 5 epochs shows very little deviation from the base model and one of those two has the highest score for every threshold and score. The model without oversampling has the lowest scores for both thresholds.

Table 10: The different scores: gilbert skill score (GSS), hit rate (H), false alarm rate (F) and bias (B) for the different models with out oversampling, with maximum oversampling 100 and 5 bins and finetuning of this model with 5, 10 or 50 epochs with a learning rate of 10^{-7} at a threshold of 60 ppb. The highest value for each score is marked in bold face.

Threshold: 60 ppb	GSS	H	F	B
5 epochs	**0.4079**	0.6574	0.0241	1.1946
10 epochs	0.4074	0.6491	0.0234	1.1693
50 epochs	0.4025	0.5979	0.0189	1.0183
rate 100, bins 5	0.4074	**0.6665**	**0.0251**	**1.2260**
no oversampling	0.3482	0.4266	0.0079	0.6031

Table 11: The different scores: gilbert skill score (GSS), hit rate (H), false alarm rate (F) and bias (B) for the different models with out oversampling, with maximum oversampling 100 and 5 bins and finetuning of this model with 5, 10 or 50 epochs with a learning rate of 10^{-7} at a threshold of 80 ppb. The highest value for each score is marked in bold face.

Threshold: 80 ppb	GSS	H	F	B
5 epochs	0.1168	0.1301	0.00037	0.2374
10 epochs	0.0974	0.1064	0.00030	0.1928
50 epochs	0.0333	0.0342	0.00007	0.0551
rate 100, bins 5	**0.1338**	**0.1520**	**0.00045**	**0.2811**
no oversampling	0.0185	0.0190	0.00006	0.0370



(a) learning rate 10^{-6}, 10 epochs

(b) learning rate 10^{-7}, 50 epochs

Figure 16: Conditional quantile plots for the finetuned models with a learning rate of 10^{-6} and 10 epochs and a learning rate 10^{-7} and 50 epochs. At the bottom the forecast distribution is displayed in a logarithmic scale and from bottom left to top right the observation distribution and its quantiles given a certain forecast value are shown.

Figure 16 shows the conditional quantile plots for the models with a learning rate of 10^{-6} and 10 epochs (fig. 16a) and a learning rate 10^{-7} and 50 epochs (fig. 16b). It underlines the similarity between the two models that was identifiable in the score plots (fig. 13, 15) and the score tables for the 60 ppb and 80 ppb thresholds (tab. 8, 9, 10, 11). The conditional quantile plots are very similar, both in the overall forecast distribution and in the course of the quantile lines. The only difference is on the right end of the plot, where the quantile lines merge only for the model with a learning rate of 10^{-6} and 10 epochs (fig. 16a). This is due to the fact that the left model has one forecast for 82 ppb and 83 ppb while the model on the right has one forecast for 84 ppb and 85 ppb and no forecasts for 83 ppb. This discontinuance causes the quantile lines to diverge while the others converge, but is no indicator for significant differences between the models.
Overall the finetuning seems to (partially) revert the changes that were made by the oversampling. Especially for the contingency score plots (fig. 13, 15) holds: The higher the learning rate or number of epochs, the closer the models get to the model without oversampling. One exception is the false alarm rate (fig. 13c, 15c) from 10 ppb to 40 ppb where the rate is higher for the models with higher learning rate or number of epochs. The conditional quantile plots (fig. 14, 16) also show that the systematic overestimation that happens for the base model gets (partially) reverted by the finetuning. But they also show that the finetuning overall decreases the range of forecasted concentrations, therefore making the forecast distribution more compact. Additionally adjustments in the number of epochs and the learning rate have a very similar effect, resulting in the model with 10 epochs and a learning rate of 10^{-6} and the model with 50 epochs but a learning rate of 10^{-7} being almost identical.

6 Discussion

The results section was split into two parts, investigating the effect of oversampling in the first part and finetuning in the second part. The discussion will follow this structure and aims on explaining the metrics and results that where highlighted in the results section as well as linking these and drawing conclusions from them.
The contingency score plots in the oversampling section (fig. 8, 9, 11) have shown a significant increase that is caused by oversampling, in the positive as well as in the negative oriented metrics. The deviation between the models only appears in a certain range, ending at 85 ppb for all scores. This can be explained with the conditional quantile plots in figures 10 and 12. The forecast distribution of the oversampled models has an increase in higher forecasts but the maximum value that is forecasted is not significantly higher. This leads to none of the models forecasting concentrations above 85 ppb which results in identical scores for all models after that threshold. The increase in the scores for all models above 85 ppb is only due to the way that the contingency tables are created as explained in section 4.4.2. The reason why the oversampling does not increase the maximum forecast might be the very low frequency of the samples that are above the maximum forecast of 85 ppb. Table 1 shows that when using 10 bins, the oversampling rates needed for the last two bins are 3300 and 15880, respectively. This results in the extreme values still being under represented in the training data (fig. 7). Another explanation might be,

that the amount of samples for the extreme values is to low for the model to learn from it, no matter how often the samples are multiplied.

When comparing the impact of the oversampling on the positive oriented metrics to the negative oriented metrics, the easiest comparison is in hit rate and false alarm rate since these metrics are a counterpart to each other and a change in them can be directly translated into the impact that this has on the forecasts of the model. Tables 2-7 that display the scores at 60 ppb and 80 ppb show that oversampling causes a higher absolute increase in the hit rate while the false alarm rate has a stronger relative increase. This trade off between hit rate and false alarm rate can also be observed in the conditional quantile plots. The increased hit rate (fig. 8b, 9b, 11b) matches to the increase in higher forecasts (fig 10, 12) while the increase in false alarm rate (fig. 8c, 9c, 11c) can also be seen at the systematic overestimation which is indicated by the 50% quantile being below the reference line for all oversampling models (fig. 12). The ranges where the false alarm rate is increased and the ranges where the oversampled models show systematic overestimation also match, both from ca. 30 ppb to 85 ppb.

Apart from the trade off between hit rate and false alarm rate, the oversampling models also show a gilbert higher skill score for the interval from 50 ppb to 85 ppb (fig. 8a, 9a, 11a). In contrast to the hit rate (eq. 14), the gilbert skill score (eq. 10) also accounts for samples that are falsely classified to be above the threshold. While the hit rate can be artificially increased by only forecasting very high concentrations, this is not possible for the gilbert skill score. Therefore, the increase in gilbert skill score leads to the conclusion that oversampling increases the ability to distinguish between below or above a threshold for thresholds between 50 ppb and 80 ppb. The increase in the hit rate between 30 and 50 ppb, where the gilbert skill score is not increased, can be purely attributed to the increase in the false alarm rate that is caused by systematic overestimation.

When comparing the different oversampling models, the tables 2-7 show, that the differences between some of the models are only marginal. While the models with the lowest or highest parameters show some stronger deviations, either with positive oriented metrics being lower or negative oriented metrics being higher, the models with oversampling rates of 50 or 100 and 5 or 10 bins (tab. 6, 7, fig. 11) have very similar scores. Overall the model with a maximum oversampling rate of 100 and 5 bins showed (one of) the highest increases in the positive oriented metrics without having significantly higher negative oriented metrics when compared to the other oversampled models. The model achieves the highest increase in the hit rate from 43% to 67% at 60 ppb and from 1.9% to 15.2% at 80 ppb while only increasing the false alarm rate from 0.8% to 2.5% at 60 ppb and not increasing it at 80 ppb. It also has the highest gilbert skill score at 80 ppb and close to the highest gilbert skill score at 60 ppb. This is why it can be considered the best model from the oversampling section and why it was used as base model for the finetuning.

For the two finetuning experiments, the learning rate and the number of epochs where used as parameters, respectively. Both experiments exhibited very consistent results both in the contingency score plots (fig. 13, 15) and the conditional quantile plots (fig. 14 and 16). This is especially highlighted by figure 16 that displays the conditional quantile plots for the model with a learning rate of 10^{-6} and 10 epochs and the model with a learning rate of 10^{-7} and 50 epochs. One having the higher learning rate and the other training for more epochs leads to them having very similar forecast and forecast observation distributions. The conclusion that can be drawn from that, is that adjustments in the learning rate or the number of epochs have a very similar effect.

The gilbert skill score (fig. 13a, 15a), the hit rate (fig. 13b, 15b) and the bias (fig. 13d, 15d) showed very similar patterns. For both experiments and all of the three scores, the model without oversampling had the lowest score while the base model had the highest score. The finetuned models where in between and the higher the learning rate or number of epochs, the closer they were to the model without finetuning. This indicates that the finetuning reverts some of the changes that are caused by the oversampling. The false alarm rate (fig. 13c, 15c) for thresholds above 40 ppb showed the same results as the other three metrics. But for thresholds between 10 ppb and 30 ppb, the finetuned models had the highest false alarm rate. Additionally an increase in learning rate or number of epochs led to an increase in the false alarm rate in this threshold range.

This is inline with the results highlighted by the conditional quantile plots in figures 14 and 16. They show that the finetuning decreases the range of values that are forecasted, leading to an increase in the lowest forecasted concentration and a decrease in the highest forecasted concentration. This leads to an increase in the false alarm rate (eq. 15) since more observations with lower concentrations are being overestimated. The conditional quantile plots in figures 14b and 14c also show that this effect is stronger for a higher learning rate which also matches the false alarm rate plots.

Overall the finetuning does not seem do add any additional value but only to revert some of the improvements that were made by the oversampling. It additionally results in a smaller range of forecasted concentrations which is counterproductive when the aim is to increase forecast precision for extreme values.

In summary, the results from Buda et al. [8] and Johnson et al. [22] that were found on extreme event forecasting cannot be fully transferred to extreme ozone concentration forecasting, as a time series regression task. The finetuning does not generate new forecasting abilities but only reverts some of the changes made by the oversampling. The oversampling partially succeeded but the cost of it is more significant than in the studies from Buda et al. and Johnson et al.. The results obtained in this study on extreme values in time series data are in between the results on extreme event forecasting, where the oversampling does not seem to have significant drawbacks, and the free lunch theorem from Wolpert et al. [54], that implies that improvement always comes at a cost. In this case this means that an increase the gilbert skill score and hit rate and number of higher concentration forecasts comes at the cost of also increasing the false alarm rate and the bias as well as having a systematic overestimation for certain concentration ranges.

7 Conclusion

The aim of the study was to investigate the effect of oversampling and oversampling with finetuning on the ability of the neural network to predict high and extreme ozone concentrations. Especially the improvements for forecasts of extreme values could not be achieved with either of the methods since they did not increase the maximum forecasted concentration significantly. The oversampling can still be seen as success for high ozone concentrations, increasing the quantity of higher forecasts and increasing gilbert skill score and hit rate for thresholds up to 85 ppb. These improvements come at the cost of also increasing false alarm rate and bias paired with a systematic overestimation in the ranges where the scores are increased. For applications where a high hit rate of threshold exceedances is important and sacrifices in the false alarm rate and the bias can be made in order to achieve it, oversampling can be a valuable method. The model with an oversampling rate of 100 and 5 bins, which brought the overall best results, achieved an increase in the hit rate from 43% to 67% at 60 ppb and from 1.8% to 15% at 80 ppb. Although this is a strong increase, for some applications these hit rates, especially at 80 ppb, might still be to low.

Further investigations on different oversampling techniques for this application might be useful. Possible changes to the oversampling method used in this study would be not using equally spaced bins or using a soft cap for the maximum oversampling rate that scales with the desired oversampling rate. Additionally the usage of completely different and more sophisticated oversampling methods like SMOTE would be conceivable. A problem that synthesizing oversampling methods like SMOTE might face, are physical constraints which the samples are subject to. Furthermore, the application of oversampling on multiple time steps forecasts would be useful for real world application and might bring different results.

Apart from that, the transfer of oversampling and finetuning to other time series forecasting tasks, especially with gaussian sample distributions, might be interesting in order to investigate whether the lack of transferability of the results from imbalanced data problems is due to the overall task of time series forecasting or rather because of the non gaussian distribution of ozone concentrations.

A Data set and List of stations

Table 12: Number of stations, total number of samples and various statistics on the number of sample per station in the training, validation and test data sets. The number of samples per station varies as not all stations have data through the full period.

	train	val	test
no. stations	318	219	213
no. samples	743247	147988	245647
mean	2337	675	1153
std	876	84	491
min	169	154	151
5%	747	516	337
10%	1037	623	354
25%	1611	680	616
50%	2658	702	1472
75%	3128	717	1523
90%	3276	717	1548
95%	3323	717	1564
max	3392	717	1578

Table 13: Number of samples (input and output pairs) per station separated by training (train), validation (val), and test data set. "—" denotes no samples in a set.

stat. ID	train	val	test
DEBB001	1730	—	—
DEBB006	1708	—	—
DEBB007	290	717	1542
DEBB009	1390	—	—
DEBB021	2687	694	1487
DEBB024	2836	—	—
DEBB028	1530	—	—
DEBB031	3037	—	—
DEBB036	962	—	—
DEBB038	1391	—	—
DEBB040	1277	—	—
DEBB042	3052	717	1410
DEBB043	2592	—	—
DEBB048	2667	717	1512
DEBB050	2520	717	—
DEBB051	970	—	—
DEBB052	338	—	—
DEBB053	2068	695	975
DEBB055	1865	717	1511
DEBB063	1362	717	1512

DEBB064	1469	717	1542
DEBB065	1385	687	1542
DEBB066	1447	717	1525
DEBB067	1428	717	1526
DEBB075	196	717	1512
DEBB082	—	611	1249
DEBB083	—	226	1506
DEBE010	2750	699	583
DEBE032	2658	678	569
DEBE034	2780	652	571
DEBE051	2771	682	577
DEBE056	2719	662	583
DEBE062	2658	658	620
DEBW004	3225	717	1537
DEBW006	2994	717	1306
DEBW007	2943	691	377
DEBW008	1106	—	—
DEBW010	3384	700	1537
DEBW013	2949	702	1492
DEBW019	3281	704	1537
DEBW020	1879	—	—
DEBW021	1919	—	—
DEBW023	2897	717	1522
DEBW024	3323	702	1537
DEBW025	1859	—	—
DEBW026	3348	717	413
DEBW027	3292	687	1521
DEBW028	1830	—	—
DEBW029	3331	717	1520
DEBW030	3276	353	—
DEBW031	3264	700	1437
DEBW032	2963	—	—
DEBW034	3392	699	377
DEBW035	2605	—	—
DEBW036	1492	—	—
DEBW037	3345	717	377
DEBW039	3288	702	1520
DEBW041	1928	—	—
DEBW042	2930	700	1500
DEBW044	1889	—	—
DEBW045	1107	—	—
DEBW046	3276	702	1507
DEBW047	1898	—	—
DEBW049	1083	—	—
DEBW050	1899	—	—

DEBW052	2937	717	1456
DEBW053	1917	—	—
DEBW054	1873	—	—
DEBW056	3244	717	1537
DEBW057	1042	—	—
DEBW059	3275	717	1523
DEBW060	1917	—	—
DEBW065	1838	—	—
DEBW072	838	—	—
DEBW076	3371	717	1253
DEBW081	2973	717	1537
DEBW084	2987	717	1482
DEBW087	3383	702	1520
DEBW094	3004	—	—
DEBW102	1344	—	—
DEBW103	2512	717	377
DEBW107	1786	703	608
DEBW110	1099	717	377
DEBW111	1075	688	376
DEBW112	618	717	1537
DEBW113	672	—	—
DEBY002	3196	717	634
DEBY004	3206	699	1488
DEBY005	3269	703	1557
DEBY013	1369	621	867
DEBY017	1577	—	—
DEBY020	3283	717	1497
DEBY031	3203	656	1521
DEBY032	3250	717	597
DEBY034	1877	—	—
DEBY039	2837	717	1503
DEBY047	2079	717	641
DEBY049	3190	678	1530
DEBY052	3193	700	1435
DEBY062	1375	686	352
DEBY072	3147	678	1492
DEBY077	1362	717	616
DEBY079	3167	717	558
DEBY081	3208	675	338
DEBY082	1900	—	—
DEBY088	3304	702	1503
DEBY089	2959	717	1526
DEBY092	891	—	—
DEBY099	1803	685	894
DEBY109	1254	702	1543

DEBY113	1339	695	1557
DEBY118	905	697	618
DEBY122	—	—	1310
DEHB001	2844	702	1069
DEHB002	2473	694	1119
DEHB003	2793	687	—
DEHB004	2756	700	1116
DEHB005	2726	667	1146
DEHE001	2924	717	1563
DEHE008	2861	699	1547
DEHE010	1939	—	—
DEHE013	—	717	1547
DEHE017	1832	—	—
DEHE018	3341	717	1578
DEHE019	2272	—	—
DEHE022	2971	717	1564
DEHE023	3260	700	363
DEHE024	3159	702	1563
DEHE025	1825	—	—
DEHE026	3140	685	1578
DEHE027	1809	—	—
DEHE028	3242	702	1549
DEHE030	3347	717	1578
DEHE032	3204	703	1563
DEHE033	2091	—	—
DEHE034	2211	—	—
DEHE039	—	—	1168
DEHE042	3238	717	1549
DEHE043	3256	717	1578
DEHE044	2809	717	1578
DEHE045	2521	687	1578
DEHE046	2489	703	1547
DEHE048	1028	—	—
DEHE050	1676	—	266
DEHE051	2302	717	616
DEHE052	2976	702	1577
DEHE058	778	717	1578
DEHE060	686	652	1559
DEHH008	2865	717	1485
DEHH021	2926	702	1440
DEHH033	2250	667	1473
DEHH047	2527	684	1484
DEHH049	2232	717	1486
DEHH050	2280	717	1475
DEMV001	1129	—	—

DEMV004	3199	717	1497
DEMV007	3307	717	1472
DEMV012	3118	702	1535
DEMV017	2879	704	1528
DEMV018	2063	706	151
DEMV019	1414	695	1535
DEMV021	596	676	1502
DEMV024	—	—	1345
DENI011	2761	507	1564
DENI016	3356	681	1472
DENI019	3184	465	—
DENI020	3253	692	1564
DENI028	3113	660	1024
DENI029	3206	717	1550
DENI031	3138	703	1513
DENI038	2868	717	1505
DENI041	3114	712	1522
DENI042	3196	665	1520
DENI043	3232	653	1499
DENI051	3208	702	1510
DENI052	3127	701	1544
DENI054	3217	703	1536
DENI058	2789	384	1485
DENI059	2656	717	1488
DENI060	2698	717	1523
DENI062	2538	652	1473
DENI063	2353	672	1499
DENI077	—	—	1536
DENW004	1351	—	—
DENW006	1285	675	1526
DENW008	2683	686	1485
DENW010	1623	—	—
DENW013	2062	—	—
DENW015	1730	—	—
DENW018	1512	—	—
DENW028	2621	—	—
DENW029	2736	—	—
DENW030	2921	575	1424
DENW036	1482	—	—
DENW038	2819	624	1533
DENW042	1464	—	—
DENW047	1548	—	—
DENW050	2777	—	—
DENW051	1496	—	—
DENW053	2779	653	1481

DENW059	2699	595	1350
DENW062	1291	—	—
DENW063	3010	—	—
DENW064	3112	541	1373
DENW065	3125	479	1388
DENW066	3018	—	—
DENW067	2672	670	1508
DENW068	3065	369	1307
DENW071	2684	702	1502
DENW078	1310	642	1568
DENW079	2282	688	1550
DENW080	2372	668	1468
DENW081	2425	610	1519
DENW094	1890	590	1534
DENW095	1788	642	1516
DENW096	668	—	—
DENW179	758	674	1520
DENW247	—	547	1474
DERP001	2975	717	340
DERP007	2965	717	354
DERP013	3129	700	337
DERP014	3284	688	354
DERP015	3072	699	324
DERP016	3219	717	354
DERP017	3234	717	339
DERP019	3209	700	318
DERP021	3310	702	310
DERP022	3310	693	332
DERP025	3204	679	354
DERP028	3033	662	335
DESH005	946	—	—
DESH006	823	—	—
DESH008	2989	717	339
DESH016	2965	686	—
DESH021	1423	—	—
DESH023	1682	717	354
DESH033	237	717	336
DESL003	3136	717	635
DESL008	1607	—	—
DESL011	3031	702	635
DESL012	—	—	635
DESL017	3014	717	620
DESL018	1641	699	635
DESL019	1337	687	604
DESN001	3204	670	1356

DESN004	3310	687	1399
DESN005	1493	—	—
DESN011	2810	672	1375
DESN012	3233	717	—
DESN014	2518	—	—
DESN017	3270	686	—
DESN019	3137	717	—
DESN024	3324	717	—
DESN028	683	—	—
DESN036	1001	—	—
DESN045	3100	717	1393
DESN050	3205	702	—
DESN051	3358	659	1375
DESN057	1800	—	—
DESN059	2821	703	1335
DESN074	2918	687	1346
DESN076	2632	713	1364
DESN079	—	—	1381
DESN085	702	154	—
DESN092	—	518	1379
DEST002	3342	717	1511
DEST005	1144	—	—
DEST011	3197	702	1525
DEST014	947	—	—
DEST022	1127	—	—
DEST025	812	—	—
DEST028	2998	—	—
DEST030	2575	—	—
DEST031	1096	—	—
DEST032	791	—	—
DEST039	3123	651	1521
DEST044	3194	701	1522
DEST050	3008	698	1496
DEST052	1783	—	—
DEST061	1131	—	—
DEST063	1558	—	—
DEST066	3266	635	1540
DEST069	2867	689	341
DEST070	1798	—	—
DEST071	806	—	—
DEST072	2914	685	—
DEST077	1586	643	1540
DEST078	3251	701	—
DEST089	2372	702	1452
DEST098	1400	620	1423

DEST104	—	—	1502
DETH005	3369	717	339
DETH009	3312	717	354
DETH013	3223	702	354
DETH016	2273	—	—
DETH018	3366	717	354
DETH020	3306	717	354
DETH024	1848	—	—
DETH025	2516	366	—
DETH026	2538	717	354
DETH027	2515	697	354
DETH036	3317	717	354
DETH040	3212	717	352
DETH041	3324	702	320
DETH042	3288	686	354
DETH060	2492	717	354
DETH061	2408	717	353
DETH086	169	335	—
DETH095	—	232	354
DETH096	—	—	164
DEUB001	2686	717	1412
DEUB003	1563	—	—
DEUB004	2786	717	1169
DEUB005	3051	717	1329
DEUB013	1044	—	—
DEUB021	511	—	—
DEUB022	556	—	—
DEUB026	1991	—	—
DEUB028	2820	592	1367
DEUB029	3002	717	1303
DEUB030	2953	699	1250
DEUB031	2008	—	—
DEUB032	1833	—	—
DEUB033	2281	—	—
DEUB034	1751	—	—
DEUB035	2198	—	—
DEUB036	571	—	—
DEUB038	1866	—	—
DEUB039	1838	—	—
DEUB040	1461	—	—
DEUB041	755	—	—
DEUB042	661	—	—
# Stations	318	219	213

B IntelliO3 architecture

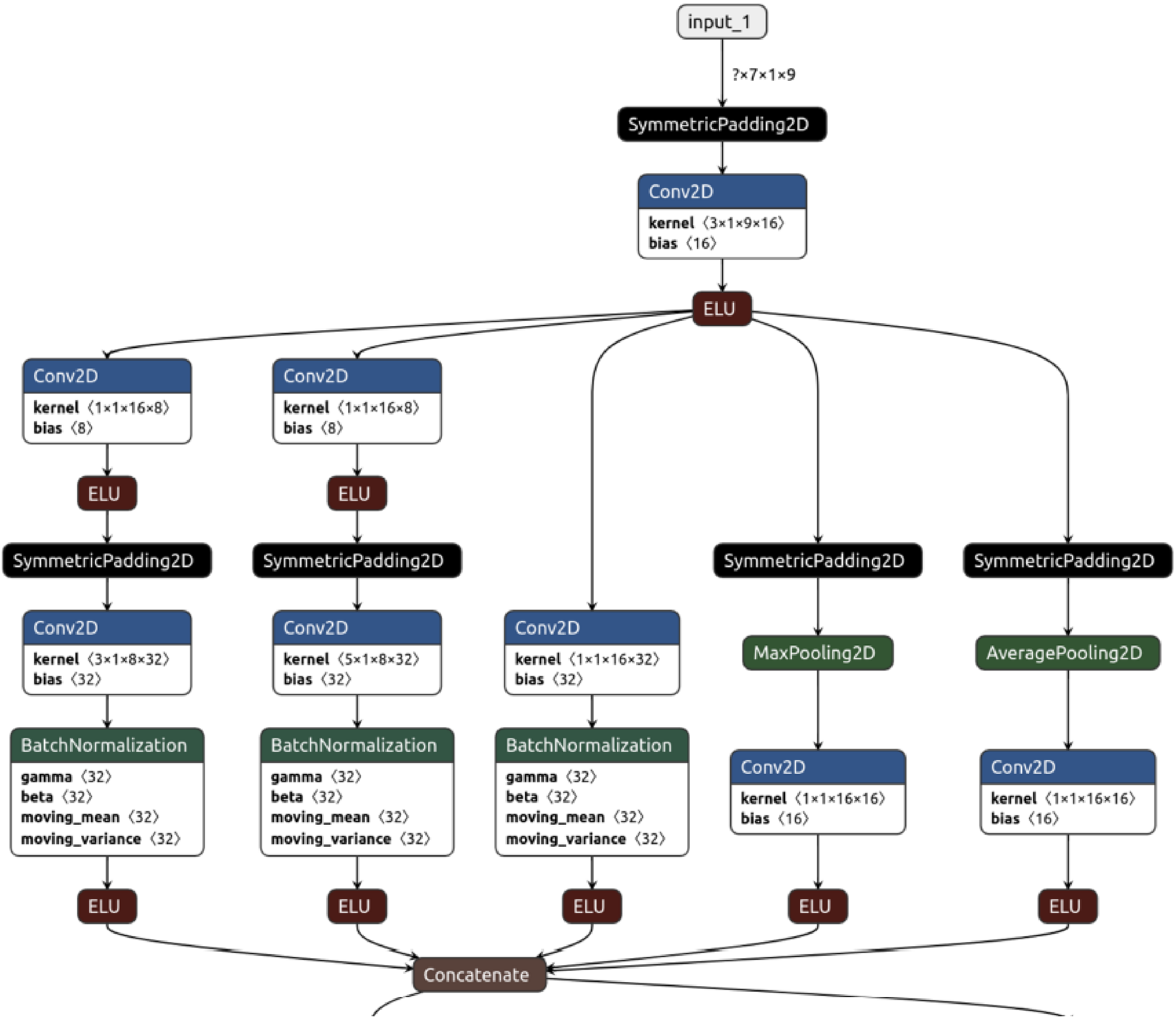


Figure 17: Figure from Kleinert et al. [26]. First part of the network architecture: The first inception block consisting of 5 parallel branches. With 5×1, 3×1, 1×1 convolutional filters, max pooling and average pooling. Padding and 1×1 filter are added on every branch to assure same output sizes.

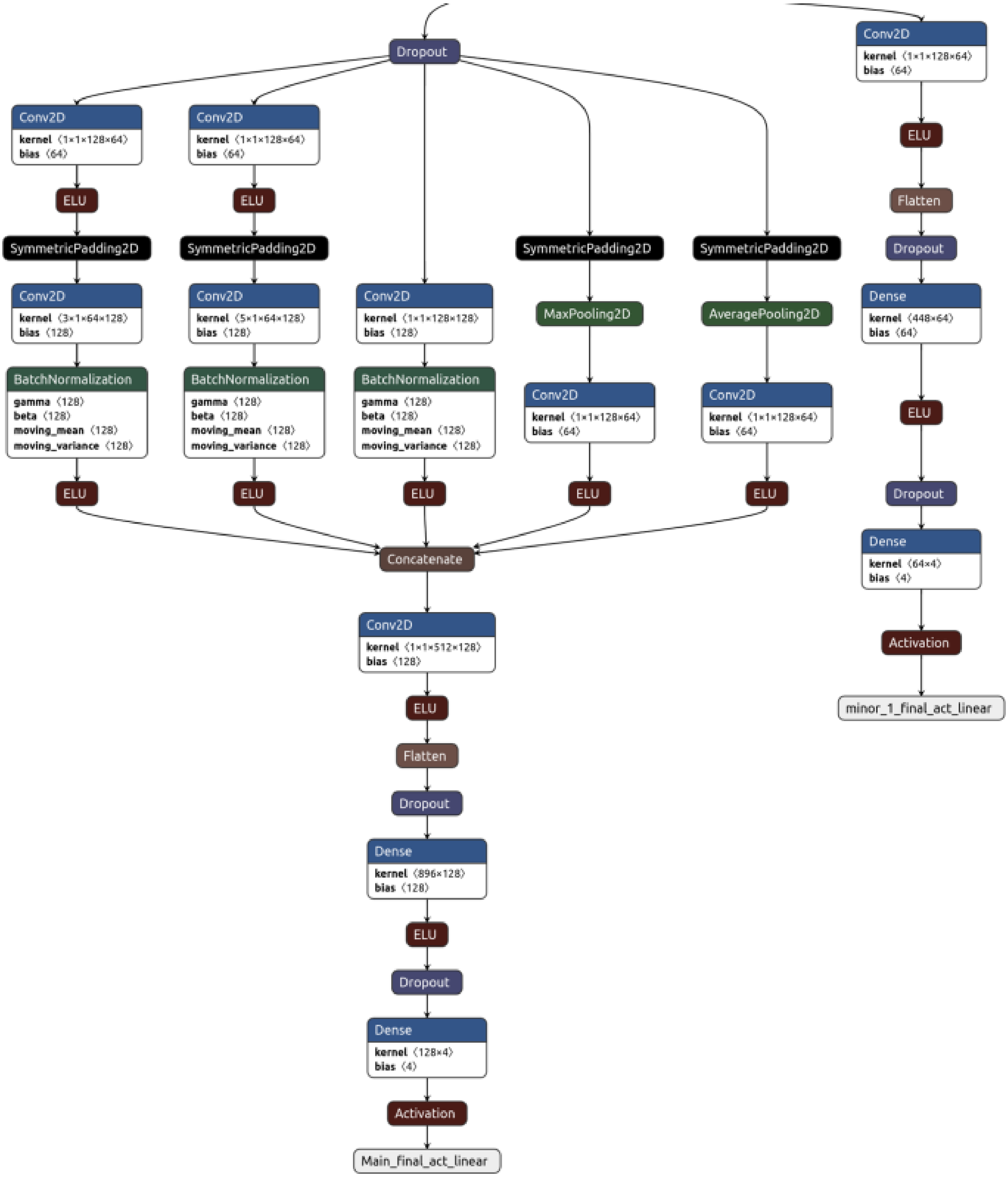


Figure 18: Figure from Kleinert et al. [26]. Second part of the network architecture: The second inception block which is identical to the first, the minor output tail on the right with two fully connected layers and the main output at the bottom with two fully connected layers.

References

[1] S. Albawi, T. A. Mohammed, and S. Al-Zawi. Understanding of a convolutional neural network. In O. Bayat, S. Aljawarneh, and H. F. Carlak, editors, *Proceedings of 2017 International Conference on Engineering & Technology (ICET'2017)*, pages 1–6, Piscataway, NJ, 2017. IEEE.

[2] L. Bai, J. Wang, X. Ma, and H. Lu. Air pollution forecasts: An overview. *International Journal of Environmental Research and Public Health*, 15(4):780, 2018.

[3] I. Basheer and M. Hajmeer. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43(1):3–31, 2000.

[4] F. Biancofiore, M. Verdecchia, P. Di Carlo, B. Tomassetti, E. Aruffo, M. Busilacchio, S. Bianco, S. Di Tommaso, and C. Colangeli. Analysis of surface ozone using a recurrent neural network. *Science of The Total Environment*, 514:379–387, 2015.

[5] C. M. Bishop. *Neural networks for pattern recognition.* Clarendon Press, Oxford, 1995.

[6] C. Bollmeyer, J. D. Keller, C. Ohlwein, S. Wahl, S. Crewell, P. Friederichs, A. Hense, J. Keune, S. Kneifel, I. Pscheidt, S. Redl, and S. Steinke. Towards a high-resolution regional reanalysis for the european cordex domain. *Quarterly Journal of the Royal Meteorological Society*, 141(686):1–15, 2015.

[7] A. Botchkarev. Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. *Interdisciplinary Journal of Information, Knowledge, and Management*, 14:045–076, 2019.

[8] M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106(7):249–259, 2018.

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[10] W. G. Cobourn, L. Dolcine, M. French, and M. C. Hubbard. A comparison of nonlinear regression and neural network models for ground-level ozone forecasting. *Journal of the Air & Waste Management Association*, 50(11):1999–2009, 2000.

[11] J. Coiffier. *Fundamentals of Numerical Weather Prediction.* Cambridge University Press, Cambridge, 2011.

[12] A. C. Comrie. Comparing neural networks and regression models for ozone forecasting. *Journal of the Air & Waste Management Association*, 47(6):653–663, 1997.

[13] David E. Rumelhart and James L. McClelland. Learning internal representations by error propagation. pages 318–362.

[14] Di Qi and A. J. Majda. Using machine learning to predict extreme events in complex systems. *Proceedings of the National Academy of Sciences*, 117(1):52–59, 2020.

[15] D. Ding, M. Zhang, X. Pan, M. Yang, and X. He. Modeling extreme events in time series prediction. In A. Teredesai, editor, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM Digital Library, pages 1114–1122, New York,NY,United States, 2019. Association for Computing Machinery.

[16] Elsevier. Statistical methods in the atmospheric sciences, volume 100 - 2nd edition, 15.06.2021.

[17] E. Eslami, Y. Choi, Y. Lops, and A. Sayeed. A real-time hourly ozone prediction system using deep convolutional neural network. *Neural Computing and Applications*, 32(13):8783–8797, 2020.

[18] F. Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. 1963.

[19] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. 2014.

[20] G. S. Handelman, H. K. Kok, R. V. Chandra, A. H. Razavi, S. Huang, M. Brooks, M. J. Lee, and H. Asadi. Peering into the black box of artificial intelligence: Evaluation metrics of machine learning methods. *AJR. American journal of roentgenology*, 212(1):38–43, 2019.

[21] T.-L. He, D. B. A. Jones, B. Huang, Y. Liu, K. Miyazaki, Z. Jiang, E. C. White, H. M. Worden, and J. R. Worden. Recurrent u-net: Deep learning to predict daily summertime ozone in the united states. 2019.

[22] J. M. Johnson and T. M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

[23] S. Khan, N. Islam, Z. Jan, I. Ud Din, and J. J. P. C. Rodrigues. A novel deep learning based framework for the detection and classification of breast cancer using transfer learning. *Pattern Recognition Letters*, 125:1–6, 2019.

[24] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization.

[25] D. P. Kingma and M. Welling. Auto-encoding variational bayes.

[26] F. Kleinert, L. H. Leufen, and M. G. Schultz. Intellio3-ts v1.0: a neural network approach to predict near-surface ozone concentrations in germany. *Geoscientific Model Development*, 14(1):1–25, 2021.

[27] J. Kukkonen, T. Olsson, D. M. Schultz, A. Baklanov, T. Klein, A. I. Miranda, A. Monteiro, M. Hirtl, V. Tarvainen, M. Boy, V.-H. Peuch, A. Poupkou, I. Kioutsioukis, S. Finardi, M. Sofiev, R. Sokhi, K. E. J. Lehtinen, K. Karatzas, R. San José, M. Astitha, G. Kallos, M. Schaap, E. Reimer, H. Jakobs, and K. Eben. A review of operational, regional-scale, chemical weather forecasting models in europe. *Atmospheric Chemistry and Physics*, 12(1):1–87, 2012.

[28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[29] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. *International Conference on Machine Learning*, pages 8093–8104, 2020.

[30] L. H. Leufen, F. Kleinert, and M. G. Schultz. Mlair (v1.0) – a tool to enable fast and flexible machine learning on air data time series. *Geoscientific Model Development*, 14(3):1553–1574, 2021.

[31] Q. Liao, M. Zhu, L. Wu, X. Pan, X. Tang, and Z. Wang. Deep learning for air quality forecasts: a review. *Current Pollution Reports*, 6(4):399–409, 2020.

[32] Y. Liu, E. Racah, Prabhat, J. Correa, A. Khosrowshahi, D. Lavers, K. Kunkel, M. Wehner, and W. Collins. Application of deep convolutional neural networks for detecting extreme weather in climate datasets. 2016.

[33] J. Ma, Z. Li, J. C. Cheng, Y. Ding, C. Lin, and Z. Xu. Air quality prediction at new stations using spatially transferred bi-directional long short-term memory network. *Science of The Total Environment*, 705:135771, 2020.

[34] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[35] Monitoring and Diagnostics, Hans-Ertel-Center fro Weather Research - Climate. Cosmo regional reanalysis - cosmo-rea6, 2021.

[36] N. Laptev, J. Yosinski, L. Li, and Slawek Smyl. Time-series extreme event forecasting with neural networks at uber. 2017.

[37] National Oceanic and Atmospheric Administration. Forecast verification glossary, 2021.

[38] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[39] S. Rich. Ozone damage to plants. *Annual Review of Phytopathology*, 2(1):253–266, 1964.

[40] S. M. Robeson and D. G. Steyn. Evaluation and comparison of statistical forecast models for daily maximum ozone concentrations. *Atmospheric Environment. Part B. Urban Atmosphere*, 24(2):303–312, 1990.

[41] A. Sayeed, Y. Choi, E. Eslami, Y. Lops, A. Roy, and J. Jung. Using a deep convolutional neural network to predict 2017 ozone concentrations, 24 hours in advance. *Neural networks : the official journal of the International Neural Network Society*, 121:396–408, 2020.

[42] J. L. Schnell, C. D. Holmes, A. Jangam, and M. J. Prather. Skill in forecasting extreme ozone pollution episodes with a global atmospheric chemistry model. *Atmospheric Chemistry and Physics*, 14(15):7721–7739, 2014.

[43] M. G. Schultz, S. Schröder, O. Lyapina, O. Cooper, I. Galbally, I. Petropavlovskikh, E. von Schneidemesser, H. Tanimoto, Y. Elshorbany, M. Naja, R. Seguel, U. Dauert, P. Eckhardt, S. Feigenspahn, M. Fiebig, A.-G. Hjellbrekke, Y.-D. Hong, P. Christian Kjeld, H. Koide, G. Lear, D. Tarasick, M. Ueno, M. Wallasch, D. Baumgardner, M.-T. Chuang, R. Gillett, M. Lee, S. Molloy, R. Moolla, T. Wang, K. Sharps, J. A. Adame, G. Ancellet, F. Apadula, P. Artaxo, M. Barlasina, M. Bogucka, P. Bonasoni, L. Chang, A. Colomb, E. Cuevas, M. Cupeiro, A. Degorska, A. Ding, M. Fröhlich, M. Frolova, H. Gadhavi, F. Gheusi, S. Gilge, M. Y. Gonzalez, V. Gros, S. H. Hamad, D. Helmig, D. Henriques, O. Hermansen, R. Holla, J. Huber, U. Im, D. A. Jaffe, N. Komala, D. Kubistin, K.-S. Lam, T. Laurila, H. Lee, I. Levy, C. Mazzoleni, L. Mazzoleni, A. McClure-Begley, M. Mohamad, M. Murovic, M. Navarro-Comas, F. Nicodim, D. Parrish, K. A. Read, N. Reid, L. Ries, P. Saxena, J. J. Schwab, Y. Scorgie, I. Senik, P. Simmonds, V. Sinha, A. Skorokhod, G. Spain, W. Spangl, R. Spoor, S. R. Springston, K. Steer, M. Steinbacher, E. Suharguniyawan, P. Torre, T. Trickl, L. Weili, R. Weller, X. Xu, L. Xue, and M. Zhiqiang. Tropospheric ozone assessment report: Database and metrics data of global surface ozone observations. *Elem Sci Anth*, 5(0):58, 2017.

[44] S. Sharma, S. Sharma, and A. Athaiya. Activation functions in neural networks. *International Journal of Engineering Applied Sciences and Technology*, 04(12):310–316, 2020.

[45] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. 2014.

[46] UCAR. Ozone in the troposphere, `https://scied.ucar.edu/learning-zone/air-quality/ozone-troposphere`, 13.07.2021.

[47] Umweltbundesamt. Ozon, `https://www.umweltbundesamt.de/themen/luft/luftschadstoffe-im-ueberblick/ozon`, 18.10.2021.

[48] Umweltbundesamt. Stationen, `https://www.umweltbundesamt.de/daten/luft/luftdaten/stationen/`, 27.08.2021.

[49] Umweltbundesamt. Überschreitungen, `https://www.umweltbundesamt.de/daten/luft/luftdaten/ueberschreitungen/`, 27.08.2021.

[50] US EPA. What is ozone?, `https://www.epa.gov/ozone-pollution-and-your-patients-health/what-ozone`, 2016.

[51] V. Prybutok, J. Yi, and David Mitchell. Comparison of neural network models with arima and regression models for prediction of houston's daily maximum ozone concentrations. *European Journal of Operational Research*, 122(1):31–40, 2000.

[52] K. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):1–40, 2016.

[53] WHO. *Health risks of air pollution in Europe-HRAPIE project: new emerging risks to health from air pollution-results from the survey of experts.* 2013.

[54] D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.

[55] X. Yu, M. O. Efe, and O. Kaynak. A general backpropagation algorithm for feedforward neural networks learning. *IEEE transactions on neural networks*, 13(1):251–254, 2002.

Eigenständigkeitserklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne die Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden. Ich versichere, dass die eingereichte elektronische Fassung der eingereichten Druckfassung vollständig entspricht.

Köln, den 2. Dezember, 2021

V. Gramlich

Vincent Gramlich