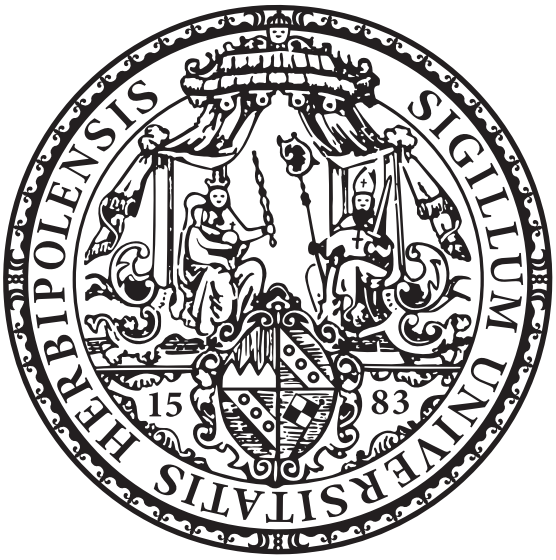
Deep Learning for Geospatial Environmental Regression

vorgelegt von

Michael Steininger

Würzburg, 2022



Kumulative Dissertation zur Erlangung des naturwissenschaftlichen Doktorgrades der Julius-Maximilians-Universität Würzburg

Abstract

Environmental issues have emerged especially since humans burned fossil fuels, which led to air pollution and climate change that harm the environment. These issues' substantial consequences evoked strong efforts towards assessing the state of our environment.

Various environmental machine learning (ML) tasks aid these efforts. These tasks concern environmental data but are common ML tasks otherwise, i.e., datasets are split (training, validatition, test), hyperparameters are optimized on validation data, and test set metrics measure a model's generalizability. This work focuses on the following environmental ML tasks: Regarding air pollution, *land use regression (LUR)* estimates air pollutant concentrations at locations where no measurements are available based on measured locations and each location's land use (e.g., industry, streets). For LUR, this work uses data from London (modeled) and Zurich (measured). Concerning climate change, a common ML task is *model output statistics (MOS)*, where a climate model's output for a study area is altered to better fit Earth observations and provide more accurate climate data. This work uses the regional climate model (RCM) REMO and Earth observations from the E-OBS dataset for MOS. Another task regarding climate is *grain size distribution interpolation* where soil properties at locations without measurements are estimated based on the few measured locations. This can provide climate models with soil information, that is important for hydrology. For this task, data from Lower Franconia is used.

Such environmental ML tasks commonly have a number of properties: (i) geospatiality, i.e., their data refers to locations relative to the Earth's surface. (ii) The environmental variables to estimate or predict are usually continuous. (iii) Data can be imbalanced due to relatively rare extreme events (e.g., extreme precipitation). (iv) Multiple related potential target variables can be available per location, since measurement devices often contain different sensors. (v) Labels are spatially often only sparsely available since conducting measurements at all locations of interest is usually infeasible. These properties present challenges but also opportunities when designing ML methods for such tasks.

In the past, environmental ML tasks have been tackled with conventional ML methods, such as linear regression or random forests (RFs). However, the field of ML has made tremendous leaps beyond these classic models through deep learning (DL). In DL, models use multiple layers of neurons, producing increasingly higher-level feature representations with growing layer depth. DL has made previously infeasible ML tasks feasible, improved the performance for many tasks in comparison to existing ML models significantly, and eliminated the need for manual feature engineering in some domains due to its ability to learn features from raw data. To harness these advantages for environmental domains it is promising to develop novel DL methods for environmental ML tasks.

This thesis presents methods for dealing with special challenges and exploiting op-

portunities inherent to environmental ML tasks in conjunction with DL. To this end, the proposed methods explore the following techniques: (i) Convolutions as in convolutional neural networks (CNNs) to exploit reoccurring spatial patterns in geospatial data. (ii) Posing the problems as regression tasks to estimate the continuous variables. (iii) Density-based weighting to improve estimation performance for rare and extreme events. (iv) Multi-task learning to make use of multiple related target variables. (v) Semi–supervised learning to cope with label sparsity. Using these techniques, this thesis considers four research questions: (i) *Can air pollution be estimated without manual feature engineering?* This is answered positively by the introduction of the CNN-based LUR model MapLUR as well as the off-the-shelf LUR solution OpenLUR. (ii) *Can colocated pollution data improve spatial air pollution models?* Multi-task learning for LUR is developed for this, showing potential for improvements with colocated data. (iii) *Can DL models improve the quality of climate model outputs?* The proposed DL climate MOS architecture ConvMOS demonstrates this. Additionally, semi-supervised training of multilayer perceptrons (MLPs) for grain size distribution interpolation is presented, which can provide improved input data. (iv) *Can DL models be taught to better estimate climate extremes?* To this end, density-based weighting for imbalanced regression (DenseLoss) is proposed and applied to the DL architecture ConvMOS, improving climate extremes estimation. These methods show how especially DL techniques can be developed for environmental ML tasks with their special characteristics in mind. This allows for better models than previously possible with conventional ML, leading to more accurate assessment and better understanding of the state of our environment.

Zusammenfassung

Umweltprobleme sind vor allem seit der Verbrennung fossiler Brennstoffe durch den Menschen entstanden. Dies hat zu Luftverschmutzung und Klimawandel geführt, was die Umwelt schädigt. Die schwerwiegenden Folgen dieser Probleme haben starke Bestrebungen ausgelöst, den Zustand unserer Umwelt zu untersuchen.

Verschiedene Ansätze des maschinellen Lernens (ML) im Umweltbereich unterstützen diese Bestrebungen. Bei diesen Aufgaben handelt es sich um gewöhnliche ML-Aufgaben, z. B. werden die Datensätze aufgeteilt (Training, Validation, Test), Hyperparameter werden auf den Validierungsdaten optimiert, und die Metriken auf den Testdaten messen die Generalisierungsfähigkeit eines Modells, aber sie befassen sich mit Umweltdaten. Diese Arbeit konzentriert sich auf die folgenden Umwelt-ML-Aufgaben: In Bezug auf Luftverschmutzung schätzt *Land Use Regression (LUR)* die Luftschadstoffkonzentration an Orten, an denen keine Messungen verfügbar sind auf Basis von gemessenen Orten und der Landnutzung (z. B. Industrie, Straßen) der Orte. Für LUR werden in dieser Arbeit Daten aus London (modelliert) und Zürich (gemessen) verwendet. Im Zusammenhang mit dem Klimawandel ist eine häufige ML-Aufgabe *Model Output Statistics (MOS)*, bei der die Ausgaben eines Klimamodells so angepasst werden, dass sie mit Erdbeobachtungen besser übereinstimmen. Dadurch werden genauere Klimadaten erzeugt. Diese Arbeit verwendet das regionale Klimamodell REMO und Erdbeobachtungen aus dem E-OBS-Datensatz für MOS. Eine weitere Aufgabe im Zusammenhang mit dem Klima ist die Interpolation von Korngrößenverteilungen. Hierbei werden Bodeneigenschaften an Orten ohne Messungen auf Basis von wenigen gemessenen Orten geschätzt, um Klimamodelle mit Bodeninformationen zu versorgen, die für die Hydrologie wichtig sind. Für diese Aufgabe werden in dieser Arbeit Bodenmessungen aus Unterfranken herangezogen.

Solche Umwelt-ML-Aufgaben haben oft eine Reihe von Eigenschaften: (i) Georäumlichkeit, d. h. ihre Daten beziehen sich auf Standorte relativ zur Erdoberfläche. (ii) Die zu schätzenden oder vorherzusagenden Umweltvariablen sind normalerweise kontinuierlich. (iii) Daten können unbalanciert sein, was auf relativ seltene Extremereignisse (z. B. extreme Niederschläge) zurückzuführen ist. (iv) Pro Standort können mehrere verwandte potenzielle Zielvariablen verfügbar sein, da Messgeräte oft verschiedene Sensoren enthalten. (v) Zielwerte sind räumlich oft nur spärlich vorhanden, da die Durchführung von Messungen an allen gewünschten Orten in der Regel nicht möglich ist. Diese Eigenschaften stellen eine Herausforderung, aber auch eine Chance bei der Entwicklung von ML-Methoden für derlei Aufgaben dar.

In der Vergangenheit wurden ML-Aufgaben im Umweltbereich mit konventionellen ML-Methoden angegangen, wie z. B. lineare Regression oder Random Forests (RFs). In den letzten Jahren hat der Bereich ML jedoch durch Deep Learning (DL) enorme Fortschritte über diese klassischen Modelle hinaus gemacht. Bei DL verwenden die Modelle mehrere

Schichten von Neuronen, die mit zunehmender Schichtungstiefe immer abstraktere Merkmalsdarstellungen erzeugen. DL hat zuvor undurchführbare ML-Aufgaben realisierbar gemacht, die Leistung für viele Aufgaben im Vergleich zu bestehenden ML-Modellen erheblich verbessert und die Notwendigkeit für manuelles Feature-Engineering in einigen Bereichen aufgrund seiner Fähigkeit, Features aus Rohdaten zu lernen, eliminiert. Um diese Vorteile für ML-Aufgaben in der Umwelt nutzbar zu machen, ist es vielversprechend, geeignete DL-Methoden für diesen Bereich zu entwickeln.

In dieser Arbeit werden Methoden zur Bewältigung der besonderen Herausforderungen und zur Nutzung der Möglichkeiten von Umwelt-ML-Aufgaben in Verbindung mit DL vorgestellt. Zu diesem Zweck werden in den vorgeschlagenen Methoden die folgenden Techniken untersucht: (i) Faltungen wie in Convolutional Neural Networks (CNNs), um wiederkehrende räumliche Muster in Geodaten zu nutzen. (ii) Probleme als Regressionsaufgaben stellen, um die kontinuierlichen Variablen zu schätzen. (iii) Dichtebasierte Gewichtung zur Verbesserung der Schätzungen bei seltenen und extremen Ereignissen. (iv) Multi-Task-Lernen, um mehrere verwandte Zielvariablen zu nutzen. (v) Halbüberwachtes Lernen, um auch mit wenigen bekannten Zielwerten zurechtzukommen. Mithilfe dieser Techniken werden in der Arbeit vier Forschungsfragen untersucht: (i) *Kann Luftverschmutzung ohne manuelles Feature Engineering geschätzt werden?* Dies wird durch die Einführung des CNN-basierten LUR-Modells MapLUR sowie der automatisierten LUR–Lösung OpenLUR positiv beantwortet. (ii) *Können kolokalisierte Verschmutzungsdaten räumliche Luftverschmutzungsmodelle verbessern?* Hierfür wird Multi-Task-Learning für LUR entwickelt, das Potenzial für Verbesserungen mit kolokalisierten Daten zeigt. (iii) *Können DL-Modelle die Qualität der Ausgaben von Klimamodellen verbessern?* Die vorgeschlagene DL-MOS-Architektur ConvMOS demonstriert das. Zusätzlich wird halbüberwachtes Training von Multilayer Perceptrons (MLPs) für die Interpolation von Korngrößenverteilungen vorgestellt, das verbesserte Eingabedaten liefern kann. (iv) *Kann man DL-Modellen beibringen, Klimaextreme besser abzuschätzen?* Zu diesem Zweck wird eine dichtebasierte Gewichtung für unbalancierte Regression (DenseLoss) vorgeschlagen und auf die DL-Architektur ConvMOS angewendet, um die Schätzung von Klimaextremen zu verbessern. Diese Methoden zeigen, wie speziell DL-Techniken für Umwelt-ML-Aufgaben unter Berücksichtigung ihrer besonderen Eigenschaften entwickelt werden können. Dies ermöglicht bessere Modelle als konventionelles ML bisher erlaubt hat, was zu einer genaueren Bewertung und einem besseren Verständnis des Zustands unserer Umwelt führt.

Acknowledgements

Completing a dissertation has been one of the most challenging endeavors I embarked upon, and as such it was impossible to accomplish this without the support of many people. Firstly, I want to thank my supervisor Andreas Hotho who gave me direction, encouragement, and also provided valuable feedback. The work at his Chair of Data Science allowed me to learn many new things, see new places (at least to the extent Covid-19 allowed), and get to know many nice and interesting people.

Many of these nice and interesting people are my colleagues at the Chair of Data Science with whom it was a pleasure to work with. We had many fruitful discussions and also many fun times on retreats and other occasions. Special thanks go out to Konstantin Kobs, Padraig Davidson, and Tobias Koopmann with whom I have spent a lot of time since my start as a PhD student. They have been very supportive and great friends. More special thanks go out to my admin-colleagues Albin Zehe, Alexander Dallmann, and Jan Pfister with whom it was almost pleasant to fix near-catastrophic failures of our computing cluster that usually occurred at non-ideal times. I also thank my former colleague Martin Becker who has been a great mentor during my time as a master student and during my start as a PhD student. Besides my colleagues at the Chair of Data Science, I also thank the people I had the pleasure of working with from the Geography department. Especially Daniel Abel and Katrin Ziegler were great collaborators who helped me considerably by demystifying climate modeling for me, without which my work on climate model output statistics (MOS) would have been tremendously more difficult.

Finally, I thank my family for supporting me all those years. I thank my wife Anika for providing me with boundless support and love which helped me especially through the tougher times that inevitably come to any PhD student. I am convinced that I could have not completed my dissertation without their support.

Michael Steininger
Würzburg, 10th October 2022

Contents

Acronyms

CEDGAN	conditional encoder-decoder generative adversarial network
CNN	convolutional neural network
DL	deep learning
DOG	Data-driven, Open, Global
FDS	feature distribution smoothing
GAM	generalized additive model
GCM	general circulation model
IDW	inverse distance weighting
KDE	kernel density estimation
kNN	k-nearest neighbor
LAEI	London Atmospheric Emissions Inventory
LDS	label distribution smoothing
LSTM	long short-term memory
LUR	land use regression
MAE	mean absolute error
ML	machine learning
MLP	multilayer perceptron
MOS	model output statistics
MSE	mean squared error
NN	neural network
OSM	OpenStreetMap
PCA	principal component analysis

PCR	principal component regression
PM	particulate matter
RCM	regional climate model
RF	random forest
RMSE	root mean squared error
SPT	standard penetration test
SVM	support vector machine
UFP	ultra-fine particle
UN	United Nations

1. Introduction

Environmental issues and their consequences have become increasingly prominent and important in recent years. People are more and more aware of the difficulties we encounter due to detrimental human influence on the environment like climate change or air pollution [48]. Because of this, scientists research and assess the state of the environment as well as the influence this state has on us humans, but also animals or plants [65].

Another development of recent years are the considerable advances made in artificial intelligence and machine learning (ML). Especially deep learning (DL) and large neural networks pushed the boundaries in many domains like computer vision [36] or natural language processing [21]. Methods from ML have long been used for environmental tasks, for example, to provide forecasts [89], estimate environmental conditions at locations where no measurements can feasibly be obtained [124], or enhance estimates from models based on differential equations [106]. These approaches often rely on conventional ML models like linear regression. However, modern and powerful ML approaches like DL may also advance the state of the art for these domains [122].

This thesis aims to bridge the gap between environmental issues and powerful, modern ML techniques like DL, as depicted in Figure 1.1. Of all the pressing environmental issues that currently exist, this work focuses on air pollution and climate change. The United Nations (UN) deem pollution and climate change "planetary crises", highlighting their importance [111]. For ML tasks in these domains, the contributions of this work show how to develop novel, powerful ML and DL models by combining techniques that fit well with the special characteristics of the respective tasks. This thesis considers four research questions — two regarding air pollution modeling and two regarding climate modeling — that are answered based on the newly proposed ML approaches.

1.1. Environmental Issues

In history, humans began to have lasting impact on the Earth's environment long ago. Starting from 10 000 to 8000 years ago, humans transformed the environment lastingly, for example, through agriculture, leading to widespread deforestation and effects such as soil erosion [137]. Some of these transformations have adverse effects on the environment. Beginning in the 19th century, humans started emitting considerable amounts of greenhouse gases leading to changes in the climate [34]. Additionally, humans also emit air pollutants that are detrimental to human health, for example, through combustion of fossil fuels [70]. While these are not the only environmental issues in existence, the UN emphasize their importance by categorizing pollution and climate change as "planetary crises" [111]. Therefore, this thesis focuses on these two issues.

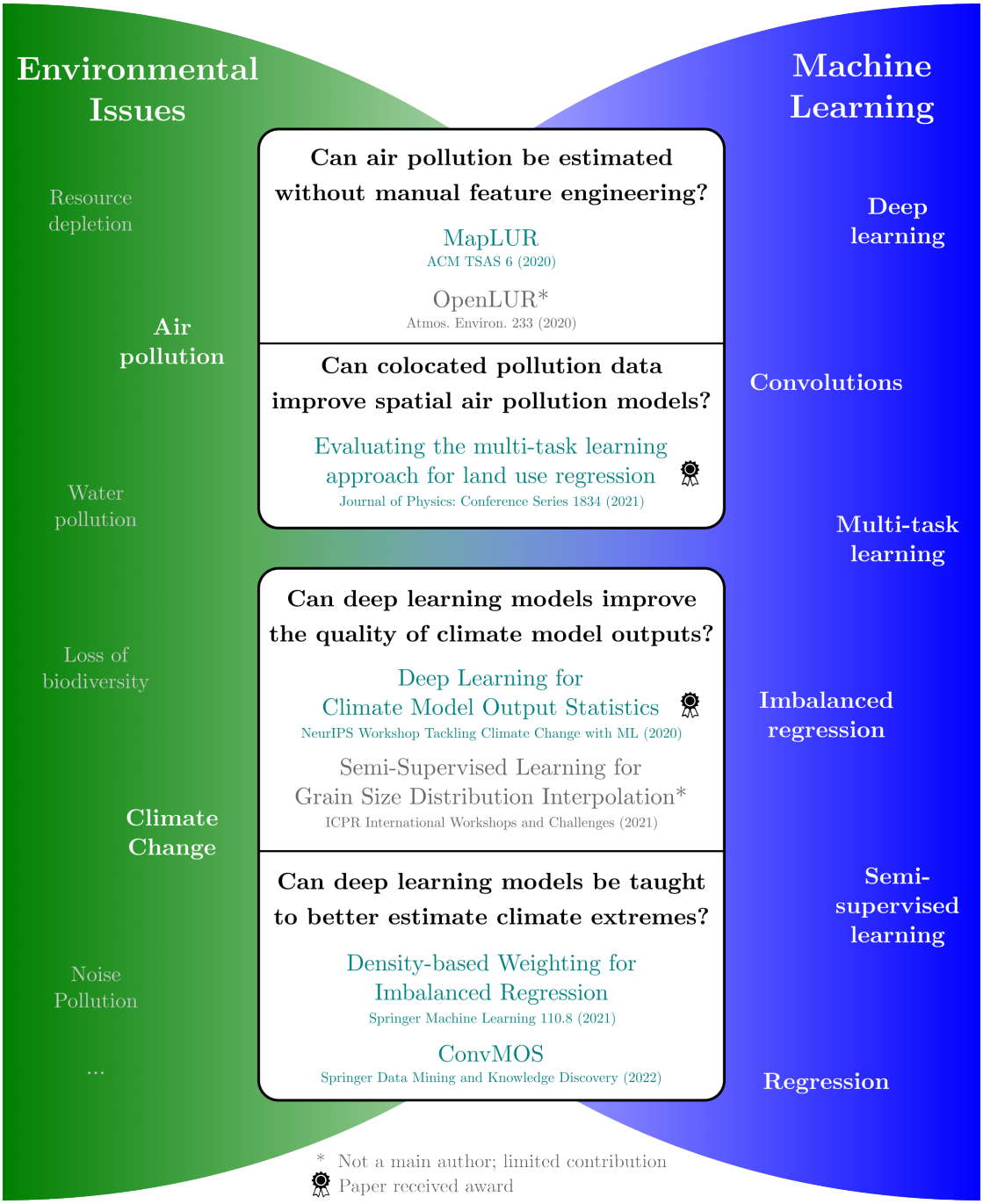


Figure 1.1.: Bridging the gap between environmental issues and ML. There are numerous environmental issues that can severely affect life on earth. At the same time, ML provides more and more powerful techniques like DL for the estimation and analysis of environmental states. This thesis aims to bridge the gap between environmental issues and ML by showing how to develop powerful, novel ML models for environmental ML tasks. Four research questions are considered that are answered based on these models.

1.1.1. Air Pollution

Air pollution has historically been a very noticeable environmental issue and still is a large problem in many parts of the world today [22]. A major source of pollutants such as particulate matter (PM), NO_2, or SO_2 is fossil fuel burning [121]. One important contributor is transportation due to the use of combustion engines but also due to PM created through brake and tire wear [77]. Other substantial contributors are heating and power generation [22].

Air pollution is known to have adverse effects on human health. Studies found that exposure to pollutants such as PM and O_3 lead to an increase in mortality [71, 141] and hospital admissions [11, 94]. Other pollutants like NO_2 or SO_2 have shown to be negatively associated with lung function [2]. In addition to influencing human health, pollutants are also known to affect plants negatively [4]. These effects make air pollution an important issue and are the reason for why there is interest in modeling air pollution concentrations.

One traditional way of modeling air pollution without ML are dispersion models. These simulate pollutant dispersion based on mathematical equations using detailed emission data of individual pollutant sources [16]. This provides information on how these individual sources affect the pollution concentrations in the surroundings, which may be of interest when approving new industrial premises, for example.

Air pollution concerns not only the emission of pollutants directly harmful to life, but it often coincides with the emission of gases that affect the climate. These greenhouse gases can increase the temperature in the atmosphere which leads to climate change, which is discussed in the following section [34].

1.1.2. Climate Change

Among the most pressing environmental issues that we encounter today is climate change. Since the late 20th century we observe a considerable increase in temperature in comparison to previous centuries. While natural variability of the climate may have some influence on this change, it is clear that the increased emissions of greenhouse gases since the industrialization provide a substantially more likely explanation for the warming [34]. Accordingly, there is a scientific consensus that the currently observed climate change is mainly caused by humans [33].

There is already evidence of observed climate change impacts today, such as changing precipitation or melting snow and ice which affects hydrological systems. Furthermore, we observe that many species show behavioral adjustments and changing abundances due to climate change. There is also evidence suggesting that climate change has negative effects on crop yield. In the future, we expect to see even more impacts, for example, increased extinction risk for many species, a decrease in food security, exacerbating existing human health problems, accelerating aggregated economic losses, and increased displacement of people. These effects show the importance of the issue and provide motivation for research on climate modeling, that allows for more accurate estimation of the climate change's development [65].

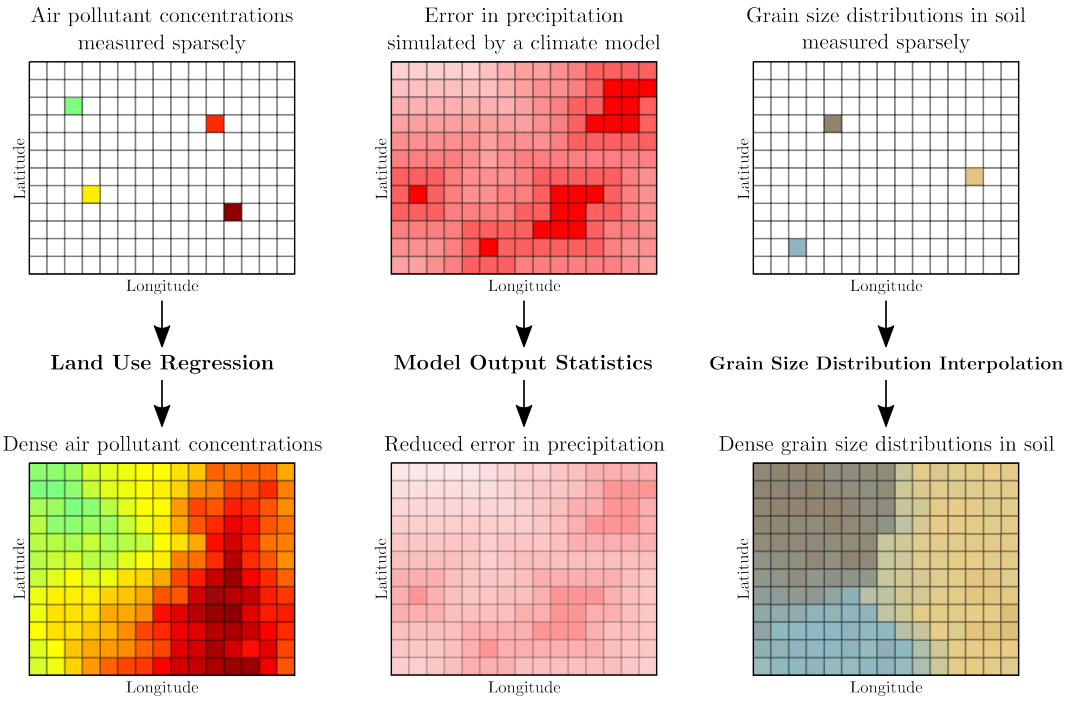
In order to assess the climate's state and make projections for potential future states, numerical climate models such as general circulation models (GCMs) and regional climate models (RCMs) are employed. These models are based on physical laws that are expressed in mathematical terms which are then implemented in a computer program. The outputs of these climate models are used to study climate change as they allow for the simulation of future climatic conditions, for example, with different greenhouse gas emission scenarios. It is possible through such simulations to assess how different future emission levels affect the climate. This information is crucial for the mitigation of climate change and the adaptation to its effects [49, 106].

1.2. Environmental ML Tasks

To assess the aforementioned environmental issues, models are built that often consist of physical laws implemented mathematically. Air pollution modeling and climate modeling are typically employed to estimate and forecast specific environmental variables of interest (e.g., air pollutant concentrations) to better grasp the severity and implications of an environmental issue. These traditional environmental modeling approaches are often supported and complemented by tasks that are solved with ML. Such ML tasks address environmental data but are otherwise technically common ML tasks, i.e., datasets are split into a training, a validation, and a test set, hyperparameters are optimized on validation set metrics, and test set metrics measure a model's generalizability. The environmental ML tasks considered in this thesis are visualized in Figure 1.2, and described in more detail in the following.

1.2.1. For Air Pollution

For air pollution, ML is often used in practice to support assessing the effects of air pollution in epidemiological studies [61]. These studies need to estimate the pollution exposure for individual participants. To this end, the studies typically use either interpolation or air pollution models such as dispersion models, which are discussed briefly in Section 1.1.1, and *land use regression (LUR)* models, as visualized in Figure 1.2a. In contrast to dispersion models, LUR models are based on ML and follow the intuition that the way we use land (e.g., for streets, industry, residential areas) impacts local pollutant concentrations. LUR models learn the relation between land use and pollutant concentrations at locations where pollutant measurements are available. These models are then applied to locations without measurements by providing their respective land use features in order to estimate local pollutant concentrations and fill the spatial gaps in pollution data [136]. LUR models are typically based on conventional ML models like linear regression [99] or random forest (RF) [9, 20, 25, 132] and help to provide local air pollution data in order to better assess the effects of air pollution on human health and the environment.



(a) LUR **(b)** Climate MOS **(c)** Grain size distribution interpolation

Figure 1.2.: Environmental ML tasks considered in this thesis. (a) Land use regression (LUR) produces spatially dense air pollutant concentrations for a gridded region based on per-location land use information. (b) Climate model output statistics (MOS) reduces errors with respect to observational data in dense, spatially gridded climate model outputs like precipitation. (c) Grain size distribution interpolation yields spatially dense grain size distributions in soil based on relatively few soil measurements.

1.2.2. For Climate Change

Climate models are valuable tools for the assessment of climate change. However, they are not perfect [92], leading to errors, and they can be partially hindered by incorrect input data. ML can support climate models to alleviate these issues, as described in the following.

Climate model output statistics (MOS). Some processes like those concerning clouds cannot be implemented directly into climate models due to their complexity or because they are too small-scale for the model's resolution. For these processes so-called parameterizations are used, which are typically simplified replacements of the real processes. These simplifications, but also issues in grid point representation for elevation, can lead to errors in the model outputs compared to observational data. To alleviate these errors it is common to use statistical methods or techniques from ML to derive empirical relationships between the model outputs and the observations. Such techniques are called *MOS* and are commonly built based on conventional ML models like linear regression [106] or RF [102, 125]. Climate MOS, as visualized in Figure 1.2b, helps to provide more accurate climate data especially at a local scale [49, 106].

Spatial Interpolation of Grain Size Distributions. ML may also help provide climate models with more accurate input data. Climate models require some information on the world as input in order to do their calculations. Measurements containing this information are often not directly available for all locations of interest. Therefore, such missing values have to be estimated from the existing measurements in the vicinity. An example for this is data on the grain size distributions in the soil, which affects hydrological processes like groundwater recharge, infiltration rates or surface flow [75]. Grain size distributions are estimated for locations without soil samples through statistical techniques like interpolation, as shown in Figure 1.2c. More accurate *grain size distribution interpolation* may help climate models to produce more accurate forecasts [75].

1.3. Common Properties of Environmental ML Tasks

ML is used in many environmental domains like air pollution modeling or climate modeling. Due to the nature of these domains, resulting environmental ML tasks have a number of typical properties. Understanding these properties is crucial for developing efficient and powerful solutions to these tasks. The following introduces common properties based on which the solutions presented in this thesis are developed.

1.3.1. Geospatiality

When measuring the state of our environment, we typically collect data through sensors which are deployed at particular locations on Earth. Because of this, environmental data is typically geospatial, meaning that it refers to locations relative to the Earth's surface [103]. This in turn also means that there are spatial relations within the data.

For example, consider sensors measuring some environmental variable (e.g., temperature) at different locations. Through the geospatial nature of the data we know that some locations are closer to each other than others and thus potentially have measurements with higher correlation in comparison.

Geospatiality may be exploited when developing ML methods by focusing on spatially close data for the estimation or prediction of an environmental variable at a particular location. This can be interpreted as a spatially-aware feature selection that reduces the amount of input features for a ML model by disregarding data that is far away and thus potentially less helpful for the estimation or prediction at hand.

1.3.2. Continuous Variables

Environmental variables are almost always continuous in nature and rarely discrete or categorical. For example, temperature, wind, air pressure, precipitation, air pollutant concentration and most other environmental variables are measured on continuous scales. Discrete or categorical environmental variables are typically derived from originally continuous variables by separating their continuous scales into bins (e.g., no rain at a precipitation of 0 mm, rain otherwise). Values of continuous variables are ordered and — similarly as with geospatial data — some values are closer to each other than others.

The continuous nature of typical environmental variables has implications on how to design ML methods for the estimation or prediction of these variables. Namely, it affects how to pose the task for ML models. Naturally, this suits itself for considering the problem a regression task in such a setting, as will be discussed in Section 1.5.2.

1.3.3. Data Imbalance

Data imbalance is prevalent in environmental variables. They often follow skewed distributions due to the relative rarity of extreme environmental conditions in comparison to more common environmental conditions. This is intuitively clear when considering, for example, precipitation. It is very common that it does not rain at any given moment in time in most parts of the world. In contrast, extreme rainfall is rare.

Data imbalance can be a hindrance for training ML models since these typically expect roughly uniform target distributions [76]. This assumption can lead to model bias where the estimation quality of samples with relatively common target values is better compared to samples with relatively rare and extreme target values [135]. This can be problematic if one is especially interested in good estimation of these infrequent samples. In particular, this is relevant for precipitation since rare extreme rainfall can have detrimental effects like floods. It can therefore be helpful to address this issue when developing environmental ML models.

1.3.4. Colocated Sensors

When building and deploying environmental sensing devices or stations such as earth observation satellites, monitoring stations or small mobile measuring devices (e.g., Zurich air pollution data [81]), it is common to include multiple sensors. Presumably, the effort

and cost of designing, building and deploying the device or station itself is often higher compared to the cost and effort of integrating additional sensors into such a design. Thus, multiple sensors are often integrated into a measuring device or station to measure more related environmental variables in addition to the main variable of interest.

Having colocated sensors is a common property of environmental ML tasks that leads to data being available for different, related aspects of the environment at the same locations. This may be exploited when developing solutions by providing ML models with the additional, related information available.

1.3.5. Spatial Label Sparsity

As mentioned before, an important source of environmental data are measuring stations that provide data for a specific location. Inherent to data from such stations is that it is typically not available at all locations of interest since it is infeasible to deploy the necessary number of stations. This is also not exclusive to data from such stations. Another example are soil samples, where it would be expensive to densely sample large areas [75]. Because of this, data for environmental ML tasks often has only sparsely available labels, at least in the spatial dimensions.

Label sparsity can be an issue for ML models and especially DL models, since these are particularly known to need sufficiently large datasets [54]. Depending on the severity of label sparsity, the concrete task, and the chosen model, it may be necessary to think of strategies that alleviate this issue.

1.4. Deep Learning

Environmental ML tasks with their aforementioned common properties have been solved with conventional ML methods before. However, while conventional ML methods are able to model many tasks well, recent years have shown that novel neural network (NN) techniques can solve tasks that were infeasible before or provide better estimation quality than prior methods [80]. The key to this success is an approach called deep learning (DL) which is an important aspect of the solutions presented in this thesis.

With conventional ML (e.g., linear regression, logistic regression, naive Bayes, support vector machine (SVM)) the success of a method depends heavily on the data representation. Raw data — especially if high dimensional — can often not be used directly as inputs for these models. Instead, researchers and practitioners need to manually design and build feature extractors based on domain knowledge that transform the raw data into a suitable data representation [80].

In contrast, DL methods are representation-learning methods which can handle raw data by learning suitable features on their own during optimization for their respective tasks. They consist of a number of layers with each layer containing artificial neurons. The neurons' outputs are fed into a non-linear activation function and are then passed to the next layer. Each consecutive layer produces higher-level feature representations and these mappings are learned automatically during optimization [80].

DL methods substantially enhance the possibilities of ML. It has allowed for record results in computer vision [36, 78, 138], natural language processing [21, 39, 130], and medicine [45, 74, 84], to name a few fields. The availability of software libraries like PyTorch [108], Tensorflow [1], and Keras [31] has eased the adoption of DL methods. There have also been first successful attempts of DL for environmental tasks, like short-term precipitation forecasting (also known as precipitation nowcasting) [7, 129] or detection of extreme weather events [113], but there are still many environmental tasks where the development of novel DL methods is promising [122]. To this end, this thesis contributes novel DL methods for the environmental ML tasks LUR, climate MOS, and grain size distribution interpolation, showing how to make use of the special properties common to environmental ML tasks when designing DL approaches.

1.5. Deep Learning for Geospatial Environmental Regression

This thesis presents ways of developing novel DL models specifically tailored for environmental ML tasks. The promise of DL methods for environmental ML tasks is their potential for improved performance as well as their capabilities for working on raw data without the need for laborious manual feature engineering [54]. This can pave the way to novel models and approaches for estimating and predicting the state of the environment. Environmental ML tasks commonly have the aforementioned properties from which follow opportunities to exploit but also challenges to overcome. To cope with these properties, this thesis proposes the use of the solutions presented in the following.

1.5.1. Convolutions

Environmental data is often geospatial as is described in Section 1.3.1 from which follows that spatially close data often has higher correlation in comparison to data that is further away. A suitable way of exploiting this property for ML models are convolutions as used in convolutional neural networks (CNNs).

1.5.1.1. What are Convolutions?

Convolutions in their general form are defined as an operation on two functions of a real-valued argument and are denoted with an asterisk. In ML these two functions are usually the input x and the weights or the kernel w [54]:

$$(x * w)(t) = \int x(a)w(t-a)da, \tag{1.1}$$

where t and a represent values in a particular domain (e.g., time for time-series data). This operation can be used to, for example, smooth input x with a weighted average over a certain span in the domain using the kernel w [54].

Since we do not have data available continuously at every point in a particular domain (e.g., at each point in time) but at discretized points in a domain instead (e.g., every second), we use the discrete convolution in practice [54]:

$$(x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a), \tag{1.2}$$

where t as well as a are integer-valued instead of real-valued and x as well as w are only defined for integer inputs.

For geospatial data the input is typically at least two-dimensional since we have a dimension for latitudes and another dimension for longitudes. This is similar to image data where there is a dimension for width and one for height. When applying a convolution to a two-dimensional input I, we commonly use a two-dimensional kernel K that also has a two-dimensional array of weights [54]:

$$(I * K)(i,j) = \sum_{m} \sum_{n} I(m,n)K(i-m, j-n). \tag{1.3}$$

1.5.1.2. Properties of Convolutions

Convolutions have a number of properties that are beneficial for geospatial data, which are described in the following.

Convolutions have *sparse interactions* since — in contrast to fully connected layers of multilayer perceptrons (MLPs) — not each input unit is connected to each output unit but instead only the input units connected to the relatively small kernel affect a particular output unit [54]. These sparse interactions are well suited for geospatial data since they exploit the relatively high correlations in spatially close data by connecting nearby input units in a kernel and omitting interactions with less related input units that are spatially further away. This reduces the number of model parameters substantially while also improving efficiency [54].

Each member of a kernel in a convolution is applied to every position of the input (disregarding potentially different behavior at boundary input units). Regardless of the position a kernel is applied to, its parameters are the same and there are thus no separate parameters to learn per location. This is called *parameter sharing* and it further improves efficiency [54].

From the convolution's parameter sharing follows *translation-equivariance*. This means that when a convolution's input is translated or shifted, its output is translated or shifted in the same way [54]. From this property in conjunction with parameter sharing follows that the kernels can detect features regardless of their specific position in the data. Geospatial data benefits from this since helpful spatial patterns only have to be learned once and not for each location individually.

This thesis shows that convolutions with their aforementioned properties are indeed well-suited for environmental ML tasks by demonstrating ways of developing novel DL models with CNNs for geospatial environmental data. To this end, this thesis proposes a novel convolutional DL model for LUR called MapLUR (see Appendix A.1) to answer the research question *Can air pollution be estimated without manual feature engineering?*. MapLUR is able to automatically learn relevant features from map images for the estimation of pollutant concentrations, while being able to achieve state-of-the-art

performance in settings with large labeled datasets [136]. Analysis shows that MapLUR automatically learns with its convolutions to detect, for example, streets in map images which it correctly associates with higher pollution concentrations [136]. Furthermore, this thesis proposes a novel convolutional DL model for climate MOS called ConvMOS (see Appendices A.3 and A.4) to answer the research question *Can DL models improve the quality of climate model outputs?*. ConvMOS is specifically tailored towards reducing errors in geospatial climate data stemming from climate models by using convolutions that are able to learn spatial error patterns as well as modules that learn error characteristics for individual locations. It provides significantly better performance than conventional ML models as well as CNNs not adapted for this specific task [133, 134].

1.5.2. Regression

Environmental variables are often continuous. This has implications on how to pose the task for ML models, since these models have to be designed and trained differently depending on whether they estimate continuous or categorical values.

The natural way of modeling the estimation or prediction of continuous variables is by posing it as a regression task. These tasks are specifically designed to use ML models to estimate or predict a numerical value. The alternative would be classification tasks that ask ML models to specify to which category a sample belongs to. However, such models would require binning the continuous variable and would also not inherently respect the order of the values and the distances to each other [54]. It is therefore sensible to develop regression models for environmental ML tasks which is why this thesis focuses on the development of such models.

1.5.3. Density-based Weighting

The imbalanced nature of environmental data can lead to non-optimal performance for the estimation or prediction of rare and extreme events, which can be of particular interest especially in the climate domain (e.g., extreme rainfall events). To alleviate this issue and answer the research question *Can DL models be taught to better estimate climate extremes?*, this thesis proposes sample weighting for regression tasks based on the target variable's density to emphasize the importance of rare samples in comparison to more common samples. This can lead to improved performance for rare and extreme samples [135].

To combat data imbalance issues there are two basic approaches: resampling and sample weighting. The former changes the data distribution by generating new data points from existing rare samples and by omitting samples with target values in relatively common target value ranges. However, the latter approach can be more desirable since omitting samples can remove useful information while generating additional data points from existing data may lead to overfitting. The sample weighting method DenseWeight and the cost-sensitive learning method DenseLoss (which is based on DenseWeight), that are proposed in this thesis, are among the first sample weighting methods for regression tasks [135].

Sample weighting for regression tasks is more complex compared to classification tasks, since there is no clear measure of rarity for an individual data point based on which one can weight its influence on training. The core idea of DenseWeight and DenseLoss is to estimate the target variable's density through kernel density estimation (KDE) in order to have a notion of rarity for each individual data point. DenseWeight's weighting function f_{w} calculates sample weights based on this rarity, so that samples with more common target values receive smaller weights than rarer samples. One can control the degree of the density-based weighting scheme with a hyperparameter $\alpha \in [0, \infty)$ ($\alpha = 0$ leads to uniform weights; larger α emphasizes density-based weighting). It is specifically designed for use in the cost-sensitive learning approach for imbalanced regression DenseLoss, where these weights are used to adjust each sample's contribution to the loss function: As such, (i) DenseWeight never assigns negative weights, avoiding maximization of the error instead of minimization during optimization. It also (ii) never assigns zero weights to avoid ignoring parts of a dataset completely. It is designed so that (iii) the mean weight over all data points is one to avoid influencing learning rates [135].

This thesis shows that the proposed method DenseLoss can reduce the estimation error of samples with rare target values better than existing resampling approaches for many tasks. Examples include statistical downscaling of precipitation, where a convolutional DL model increases the resolution of geospatial precipitation data (see Appendix A.5), and the proposed climate MOS convolutional DL architecture ConvMOS, which aims to reduce errors in geospatial precipitation data from climate models (see Appendix A.4) [133, 135].

1.5.4. Multi-Task Learning

The prevalence of colocated sensors leads to the availability of multiple related environmental variables at the same locations in environmental data, especially in air pollution data. To answer the research question *Can colocated pollution data improve spatial air pollution models?*, this thesis proposes the use of multi-task learning to exploit this circumstance and also alleviate label sparsity that is common in air pollution data. This approach is evaluated for the environmental ML task LUR [41].

In multi-task learning, a ML model is trained to correctly estimate or predict multiple target variables instead of just one. This can allow the model to leverage information from several related tasks in order to train more accurate models. As such it can be seen as an approach for ML models to mimic human learning where humans often profit from knowledge of one task when learning another related task. Multi-task learning can be especially beneficial in settings where relatively few labeled data instances are available [160].

An environmental ML task for which it is not uncommon to have multiple related potential target variables available for the same locations due to colocated sensors is LUR. In this task, a model learns to estimate pollutant concentrations at locations where no measurements are available based on relatively few locations with measurements. This thesis answers the research question *Can colocated pollution data improve spatial air pollution models?* by evaluating multi-task learning for LUR with MLPs (see

Appendix A.2). Results show that multi-task learning can indeed improve model performance significantly but only given sufficient data. This suggests that it is not suitable for lowering a model's data requirements, but it can optimize estimation quality to a certain extent [41].

1.5.5. Semi-Supervised Learning

One common property for environmental ML tasks is spatial label sparsity, as discussed in Section 1.3.5. One potential strategy to cope with this is semi-supervised learning. In a work that I have contributed to, the use of semi-supervised learning in environmental ML tasks with label sparsity is proposed and evaluated for grain size distribution interpolation with MLPs [75].

In semi-supervised learning not only samples with labels are used for model training but also samples without labels. This combination of supervised learning based on labeled samples and unsupervised learning based on unlabeled samples can help improve model performance and is especially relevant when there are relatively few labeled samples [143].

In the proposed model for grain size distribution interpolation (see Appendix B.2), semi-supervised learning helps to achieve better results in comparison to supervised methods [75]. This can provide climate models with more accurate input data, potentially leading to more accurate simulations and thus presenting another answer to this thesis' research question *Can DL models improve the quality of climate model outputs?*.

1.6. Contributions

This thesis contributes novel methods for environmental ML tasks and shows how to design effective, modern ML approaches with techniques such as DL that are suitable for the typical properties of environmental data. The proposed approaches consider either air pollution or climate data and aim to answer research questions, as presented in the following.

1.6.1. Air Pollution

In air pollution modeling, conventional ML models that aim to estimate a single pollutant based on manually engineered features are prevalent for solving the LUR task. To push the boundaries of what is possible in this domain with modern ML approaches, this thesis considers the following research questions.

1.6.1.1. Can Air Pollution be Estimated Without Manual Feature Engineering?

Prior work on LUR typically employed conventional ML models like linear regression and created features through considerable manual work for these models. The features are often calculated from at least partially non-public data that is elaborately gathered from different sources [15, 57]. To potentially ease applicability of LUR models, it is interesting to look for ways that avoid this laborious process.

One answer to this question can be DL models, since these are known to train on raw data in an end-to-end manner, meaning that they do not necessarily need tailor-made features. Instead, they can extract useful features from raw data themselves, potentially eliminating the need for manual feature engineering [80]. This thesis shows how to develop an effective, novel DL model with the geospatiality of air pollution data in mind by presenting *MapLUR* (see Appendix A.1, [136]). MapLUR is a DL model based on convolutions for LUR that learns to estimate pollutant concentrations based on map images and/or satellite images. The work on MapLUR also proposes the *Data-driven, Open, Global (DOG)* paradigm for LUR which argues for the automatic extraction of features (e.g., through DL) from openly and globally available data. MapLUR is able to outperform existing conventional LUR approaches on modeled NO_2 concentrations in Central London, showing that this approach can indeed work.

In another work, to which I provided only limited contribution, the question was answered differently, by building a novel LUR approach called *OpenLUR* (see Appendix B.1, [79]) that derives many features from OpenStreetMap (OSM) [104] and applies ML methods which allow for automated hyperparameter tuning like AutoML [47] on these features. This alternative approach provides an off-the-shelf solution for LUR that also does not require laborious manual feature engineering and manual hyperparameter optimization. OpenLUR can be used to apply DL models, but it can also make use of conventional ML models.

1.6.1.2. Can Colocated Pollution Data Improve Spatial Air Pollution Models?

The devices used to measure air pollutant concentration often record not only concentrations of one pollutant but multiple ones. These colocated sensors provide multiple related potential target variables for an environmental ML model. It is therefore interesting to consider, whether this can be of use for training models.

To this end, this thesis contributes *multi-task learning for LUR* (see Appendix A.2, [41]), where a MLP is trained to estimate pollution data from multiple colocated sensors of different pollutants. The idea is that estimating concentrations of multiple pollutants at once may improve estimation quality as it effectively adds more related labeled data for the available locations. Results show that this can indeed improve model performance in certain settings. Furthermore, this approach could hypothetically reduce the effective need of labeled samples for ML models to alleviate spatial label sparsity, but the results suggest that multi-task learning for LUR can mainly improve model performance in settings with relatively large datasets and not reduce a model's data requirements. The corresponding paper to this work won the "Best Student Paper" award at the second International Conference on Frontiers of Artificial Intelligence and Machine Learning (FAIML) [41].

1.6.2. Climate

Similarly, as in the air pollution modeling domain, it is also common to use conventional ML models like linear regression for tasks like climate MOS, where a climate model's

outputs are adjusted with ML approaches to better fit with observational data [43]. To evaluate the development and application of DL-based approaches, this thesis considers the following research questions.

1.6.2.1. Can Deep Learning Models Improve the Quality of Climate Model Outputs?

Climate model outputs follow observations reasonably well in many cases, but there are still errors, especially when it comes to precipitation [106]. While conventional ML models are able to reduce these errors, recent advances in ML beg the question whether modern DL methods are able to do the same and also to which extent.

To this end, this thesis presents a novel DL architecture based on convolutions for climate MOS called *ConvMOS* (see Appendices A.3 and A.4, [133, 134]) that learns to reduce errors in climate model outputs of precipitation in order to achieve better correspondence with observational data. The CNN-based architecture is designed specifically with the errors from climate models in mind. It consists of modules with per-location parameters and modules with global parameters. The former can learn adaptations specific to each location, while the latter efficiently learn spatial error patterns due to the convolution's translation-equivariance. Results show better model performance for ConvMOS in comparison to conventional climate MOS models. The second more comprehensive follow-up paper also shows improvements in comparison to standard CNN architectures, which shows the benefits of the task specific design of the architecture [133]. The corresponding first paper won the "Best ML Innovation" award at NeurIPS 2020's "Tackling Climate Change with Machine Learning" workshop.

Another approach for more indirectly improving the quality of climate model outputs with DL models is presented in work to which I contributed but only in a limited manner. It presents a novel DL training strategy based on a MLP and *semi-supervised learning for spatial grain size distribution interpolation* (see Appendix B.2, [75]) which aims to provide more accurate soil data that can then be used in climate models for aspects like soil-hydrological processes. In the proposed approach a MLP is pre-trained on weak labels for data without measurements and fine-tuned on data with measurements to estimate the portions of clay, silt, and sand in the soil. The MLP is furthermore provided with additional information on all locations of interest like the topography. The proposed approach reduces errors in soil data in comparison to conventional interpolation techniques and plain supervised MLP training, providing more accurate soil data which is likely to provide even more accurate climate data when used in a climate model.

1.6.2.2. Can Deep Learning Models be Taught to Better Estimate Climate Extremes?

Besides mean variables, it is also of particular interest in the climate domain to look at climate extremes because of their often severe consequences. Estimating such extreme values poses a challenge to ML approaches in general but also to DL approaches since they tend to provide more conservative estimates biased towards the mean [135]. Thus,

it is interesting to consider whether DL models can be taught to better estimate climate extremes.

At the core, this is a question of learning with imbalanced data since the climate extremes can be considered rare outliers which skew the distribution with regard to the large number of common data points closer to the mean. To cope with this in a regression setting, this thesis proposes novel density-based weighting techniques for data imbalance with the sample weighting approach *DenseWeight* and the cost-sensitive learning method for NNs *DenseLoss* which is based on DenseWeight (see Appendix A.5, [135]). These methods can be used to reduce errors in the estimates of rare events like extreme precipitation, as shown in the corresponding publication. These techniques are particularly useful for environmental ML tasks due to the prevalence of data imbalance in environmental data and the importance of rare and extreme events, especially in the climate domain. In the second publication regarding ConvMOS (see Appendix A.4), DenseLoss was applied to ConvMOS, showing it allows for the improved estimation of rare extreme precipitation events.

1.6.3. Conclusion

The aforementioned contributions demonstrate for the environmental ML tasks LUR, climate MOS, and grain size interpolation how to design novel ML and especially DL approaches. These approaches make use of techniques like convolutions, regression modeling, density-based weighting, multi-task learning, and semi-supervised learning where suitable, in order to alleviate issues and grasp chances that the properties, which environmental ML tasks commonly have, provide. This thesis uses these approaches to answer research questions regarding advanced ML, air pollution modeling, and climate modeling. Improvements in environmental ML tasks can help better monitor, assess, and forecast the state of the environment, which is especially important with regard to understanding, mitigating, and adapting to the pressing environmental issues we encounter today, like air pollution and climate change.

2. Contextualization of the Contributions within Related Work

The following contextualizes the contributions of this thesis within related work in their specific field and with regard to the research question each contribution aims to answer. First, air pollution modelling or, more specifically, land use regression (LUR) is considered by reflecting on work related to MapLUR (see Appendix A.1, [136]) and OpenLUR (see Appendix B.1, [79]) with regard to the question *Can air pollution be estimated without manual feature engineering?* in Section 2.1.1. Then, multi-task learning for LUR (see Appendix A.2, [41]) is contextualized within related work for the question *Can colocated pollution data improve spatial air pollution models?* in Section 2.1.2. Next, the contributions regarding climate modeling are considered. Related work for climate model output statistics (MOS) and ConvMOS (see Appendices A.3 and A.4, [133, 134]) as well as related work for grain size distribution interpolation and the semi-supervised approach presented in this thesis (see Appendix B.2, [75]) are examined with regard to the question *Can deep learning (DL) models improve the quality of climate model outputs?* in Section 2.2.1. Finally, the question *Can DL models be taught to better estimate climate extremes?* is considered in Section 2.2.2, contextualizing the imbalanced regression contributions DenseWeight and DenseLoss (see Appendix A.5, [135]) as well as their application to ConvMOS for better estimation of climate extremes (see Appendix A.4, [133]).

2.1. Air Pollution

Modeling air pollution geospatially with LUR is an important environmental machine learning (ML) task, especially for epidemiological studies [124]. To further advance the field, this thesis considers two research questions that are answered based on the presented contributions. These contributions are contextualized within related work in the following with regard to their research question.

2.1.1. Can Air Pollution be Estimated Without Manual Feature Engineering?

Typical LUR approaches rely on laborious manual feature engineering based on the data that is available for the considered study area. In order to ease the applicability of LUR, it is interesting to consider ways of automating this process. The following presents prior work on LUR, two solutions that alleviate the need for manual feature engineering in

this domain, namely MapLUR [136] and OpenLUR [79], as well as recent developments in related work regarding LUR.

2.1.1.1. Prior Related Work

Traditionally, LUR has mostly been solved with linear regression [15, 100, 126, 154]. Important work for this was done by the Escape project [15, 44] which has established a procedure for model building that has been widely used thereafter [95, 98, 100, 147, 151]. This relatively simple approach is often suitable since there are often only few locations (typically well under 100) with measurements per city available [15]. In such settings there are typically only very sparsely available labels, which may then result in difficulties when trying to train more complex models. Nonetheless, some more complex ML models have been used, although in part by creating larger datasets with mobile measurement campaigns yielding more than 100 measured locations [57]. Random forests (RFs) have been applied successfully [19, 20, 146]. Other works also used multilayer perceptrons (MLPs) [3, 8, 23, 25, 88]. These approaches rely on manual feature engineering based on data that is often at least partially not openly available. This thesis presents two LUR approaches which aim to automate the process and focus on openly available data, namely MapLUR (see Appendix A.1, [136]) and OpenLUR (see Appendix B.1, [79]).

2.1.1.2. MapLUR

MapLUR is a novel convolutional neural network (CNN)-based DL model for LUR that automatically learns to extract suitable land use features from openly available map and/or satellite images for the estimation of pollutant concentrations. It follows the *Data-driven, Open, Global (DOG)* paradigm proposed in this contribution which stands for the automatic extraction of features from openly and globally available data. At the core, the MapLUR model consists of 15 consecutive feature-learning blocks. Each block contains a convolution layer, batch normalization [64], and ReLU activation [101]. Pooling layers are also incorporated in the first, third, fifth, seventh, tenth and thirteenth block. These are able to learn spatial features from the input images through which they implicitly recognize the relation between different types of land use (e.g., streets, industrial areas, etc.) and air pollution. The corresponding paper "MapLUR: Exploring a new paradigm for estimating air pollution using deep learning on map images" (see Appendix A.1, [136]) was published in 2020 in the ACM journal Transactions on Spatial Algorithms and Systems in its special issue on DL.

Since DL models typically require more data than conventional ML methods that require manually engineered features, MapLUR's experimental evaluation makes use of a relatively large dataset of modeled annual mean NO_2 concentrations for Central London from the London Atmospheric Emissions Inventory (LAEI) [6]. The training set obtained from LAEI data contains pollutant concentrations for 3000 locations, which also serve as validation data for hyperparameter tuning through ten-fold cross-validations. The test set contains 1500 unseen locations. For each model 40 instances were trained and tested. The reported metrics are the mean over the 40 runs. To compare MapLUR to

Table 2.1.: Results of MapLUR experiments with different input images and baseline methods on the Central London NO_2 dataset. R^2 aims to measure the portion of the target's variation that is explained by the model (1 is perfect and 0 is the result when always estimating the test data's mean). RMSE is the square root of the mean squared error (lower is better). Baseline models are implemented following LUR approaches presented in prior work. The linear regression model follows Eeftens et al. [44], the multilayer perceptron follows Alam and McNabola [8], and the random forest follows Brokamp et al. [20]. Best values are written in bold. [136]

Model	R^2	RMSE [$\mu g\, m^{-3}$]
Mean baseline	0.000	13.971
Linear regression	0.487	10.004
Multilayer Perceptron	0.499	9.887
Random Forest	0.662	8.119
MapLUR exp. 1: OpenStreetMap	**0.673**	**8.002**
MapLUR exp. 2: Google Maps	0.537	8.918
MapLUR exp. 3: Google Maps Satellite	0.206	12.389
MapLUR exp. 4: OpenStreetMap and Google Maps Satellite	0.660	8.112

commonly used LUR models, linear regression, RFs, and MLPs are trained and evaluated in addition to a simple mean baseline. Experimental results (see Table 2.1) show that MapLUR using OpenStreetMap (OSM) images as input can significantly outperform the conventional baseline ML models on this modeled data without the need for manually engineered features, proofing that MapLUR's concept is feasible and promising. Satellite imagery can also be used but results in lower performance, likely due to the noise inherent to such imagery (e.g., visible pedestrians or cars that may not be representative of traffic in general). The results suggest that DL-based LUR models do indeed have potential for air pollution modeling and provide a new approach that can make effective use of relatively large datasets by having the potential to provide improved performance while also not requiring manual feature engineering. Thus, MapLUR expands the LUR toolkit by being a method ideal for datasets containing measured locations in the thousands, while conventional ML models like RFs and MLPs are the likely tool of choice for datasets containing hundreds of measured locations, and simple linear regression models may still be ideal for datasets containing only a few dozen measured locations. Lowering these data requirement boundaries of methods that are more complex than linear regression is an interesting research direction, for which this thesis evaluates an approach that makes use of colocated pollution data in Section 2.1.2 [136].

There have been other CNN-based air pollution modeling approaches prior to the publication of MapLUR and also thereafter, but these considered air haze level estimation based on camera photos [85, 159] or air pollution prediction based on camera photos and weather information [69] instead of LUR. Ghahremanloo et al. [52] also propose a CNN-based DL model for NO_2 estimation, but this approach relies on manually

engineered features instead of automatically learning features from, for example, map images. Published concurrently with our work on MapLUR, Zheng et al. [161] estimate $PM_{2.5}$ from satellite images with a CNN and a RF, but they also use manually engineered meteorological features for their pollution estimation. In contrast to MapLUR, they furthermore focus more on modeling the temporal variability than the relation between land use and long-term mean pollutant concentrations by considering daily satellite imagery from micro-satellites with daily $PM_{2.5}$ measurements of the same locations. DL approaches are more common for temporal air pollution or air quality models, where time-series data is considered, e.g., for forecasting [10, 28, 30, 46, 51, 82, 155, 156]. In addition to being temporal models, these approaches rely on manually engineered features from various sources that are partially not publically available. Jiang et al. [68] recently proposed another CNN for estimating $PM_{2.5}$ that also focuses on temporal variability using daily satellite imagery and used a constrastive learning pre-training task. Despite it being used in a different setting (i.e., temporal daily pollutant concentrations instead of long-term mean concentrations), it also uses only imagery through which it provides another air pollution modeling technique that avoids manual feature engineering. Thus, MapLUR is still among the only CNN-based DL LUR models that can estimate air pollution without manual feature engineering.

2.1.1.3. OpenLUR

OpenLUR is a LUR approach that generates openly and globally available features from OSM [104], applies a multitude of ML models on these features, and automatically tunes their hyperparameters in order to provide an easy-to-use off-the-shelf approach to LUR. Similar as with MapLUR, OpenLUR also alleviates the need for manual feature engineering as it automatically generates suitable features. The generated features for each location of interest include the area of industrial, commercial, and residential zones in the vicinity (50 m to 3000 m away), the length of larger and smaller streets nearby, and the distances to the next motorway, primary road, traffic signal, and industrial area from the considered location. These features are then used with ML approaches like AutoML [47], RF [18], RFOstochastic which is a RF whose hyperparameters are optimized with stochastic search, and generalized additive models (GAMs), but other techniques like DL models, for example, can also be used. This approach called OpenLUR is an easy-to-use off-the-shelf approach to LUR. The corresponding paper "OpenLUR: Off-the-shelf air pollution modeling with open features and machine learning" (see Appendix B.1, [79]) was published in 2020 in the journal Atmospheric Environment. Note that I was involved with this work only as a co-author and thus, provided only limited contribution to the work.

OpenLUR is evaluated using the OpenSense dataset containing ultra-fine particle (UFP) measurements from a mobile measuring campaign in Zurich that attached sensors on top of tram cars [57]. There are 200 locations with measurements that are deemed reliable enough to calculate usable mean concentrations for three-month periods (seasons). For the experiments, this data is randomly split into ten subsets, where one subset is used for testing and the other nine subsets are used for building models. This ten-

Table 2.2.: Absolute performance gain of the models AutoML, RFOstochastic, RF, and GAM trained with OpenLUR's generated open features (Ours) over the manually engineered OpenSense features from local or closed sources with regard to RMSE and R^2. Negative (positive) RMSE (R^2) values show a better model performance when using our automatically generated and openly available OSM features and are highlighted in gray. In nearly all cases, our OSM features yield significantly better air quality predictions compared to the manually engineered OpenSense features (bold values indicate statistical significance based on a Wilcoxon signed-rank test [150] with significance level 0.05), showing that OpenLUR can alleviate the need for manual feature engineering in LUR. [79]

Season	AutoML	RFOstochastic	RF	GAM
	RMSE $[\frac{10^9 \text{particles}}{\text{m}^3}]$			
1	**−0.18**	**−0.24**	**−0.24**	**−0.30**
2	**−0.06**	**−0.08**	**−0.07**	**−0.07**
3	−0.05	0.06	**0.06**	**0.10**
4	**−0.34**	**−0.39**	**−0.40**	**−0.19**
	R^2			
1	**0.11**	**0.16**	**0.16**	**0.18**
2	**0.04**	**0.05**	**0.06**	**0.04**
3	0.03	−0.02	**−0.06**	**−0.06**
4	**0.13**	**0.19**	**0.19**	**0.03**

fold cross-validation is done 40 times per model and mean results of these 40 runs are reported. Results in Table 2.2 show that the generated open features from OSM data can even provide better performance than comparable features from local or closed sources in 3 of the 4 seasons available in the dataset (root mean squared error (RMSE) reductions between 0.05 $\frac{10^9 \text{particles}}{\text{m}^3}$ and 0.40 $\frac{10^9 \text{particles}}{\text{m}^3}$, R^2 increases between 0.03 and 0.19). Experiments also suggest that there is a tendency towards better performance for models with automatic hyperparameter optimization like AutoML and RF with stochastic hyperparameter optimization (RFOstochastic) in comparison to laborious manual optimization efforts. The open and globally available features also allow for cross-learning across data from different cities in order to potentially improve performance for a city with few data points by adding samples from another city with more measurements of the same (or at least a related) air pollutant. This may lower the data requirement boundaries of more complex LUR methods. Just as with MapLUR's evaluation (see Section 2.1.1.2), modeled air pollutant concentration data for Central London from LAEI is used in order to have data for another city besides Zurich. Results showed to improve R^2 for AutoML by 0.29 and RFOstochastic by 0.21 when training on only 20 samples from Central London in conjunction with 180 samples from Zurich in order to estimate air pollution in Central London. However, there are strongly diminishing returns as soon as there are 40 to 60 measured locations. When there are even more measured locations available, cross-learning seems to not improve performance [79].

Both before and after publication of OpenLUR, other easy-to-use software packages were released for LUR.

The package RLUR [99] is based on the R programming language [112] and provides a graphical user-interface for LUR. RLUR automatically generates features from data provided by the user, which is similar to OpenLUR and also alleviates the need for manual feature engineering. However, RLUR only considers simple linear regression models while this thesis' contribution considers more advanced models with automatic hyperparameter optimization.

PyLUR [90] is a LUR software based on Python [144]. It is similar to RLUR in that it can generate features and solely uses linear regression models, but the authors claim superior efficiency, stability, and model performance.

XLUR [97] is also a LUR tool based on Python [144]. As with RLUR and PyLUR, this software generates features and employs linear regression models, but it is integrated into the ArcGIS Pro Software [117].

Considering these other LUR tools, OpenLUR provides a unique solution to LUR with more advanced ML methods than available in comparable approaches while also automatically generating features, which alleviates the need for manual feature engineering.

2.1.1.4. Recent Related Work

Since the publication of these contributions, there are no works that I am aware of that consider eliminating manual feature engineering from LUR. However, there have been other interesting advances in LUR. One example is hybrid LUR based on both modeled pollution concentrations and measurements, which provides an interesting new direction

for model development in this setting [142]. LUR has also been used to model residential indoor air pollution, which may also be a useful direction for health studies based on such estimates [87]. There are also new findings regarding the choice of ML model, with a number of recent publications suggesting that XGBoost [29] models are also suitable for LUR [62, 152, 153].

2.1.2. Can Colocated Pollution Data Improve Spatial Air Pollution Models?

When air pollution concentrations are measured it is common to not only consider a single pollutant. Because of this, datasets can contain colocated measurements of different pollutants that may be used to facilitate ML model training for LUR through multi-task learning. The following describes prior related work regarding LUR and multi-task learning. Thereafter, this thesis' contribution of multi-task learning for LUR is presented and the usage of colocated sensor data for LUR in recent related work is addressed.

2.1.2.1. Prior Related Work

Existing LUR approaches consider one particular air pollutant at a time and train a conventional ML model to estimate this pollutant based on the features available [15, 100, 154]. However, air pollution datasets like the OpenSense dataset from Zurich [57] or the LAEI [6] tend to contain data for different pollutants at the same locations. Therefore, this work asks whether this additional colocated air pollution data can be used to improve air pollution models.

Multi-task learning has shown to improve models in numerous domains. Caruana [24] compared multi-task models and single-task models for autonomous driving tasks, image recognition tasks, and medical tasks, finding that multi-task learning improves model performance. In natural language processing, Collobert and Weston [32] proposed already in 2008 training neural networks (NNs) to solve multiple speech-related predictions simultaneously. Nowadays, large language models are known to be unsupervised multi-task learners since they are able to solve different tasks than the one they were trained in a zero-shot setting [114]. Multi-task learning has also been used successfully in computer vision tasks, like anomaly detection in railway tracks [53] or semantic and instance segmentation [72]. Drug discovery models also benefited from this paradigm, showing increased accuracy and learned shared representations useful for other tasks for which the model was not trained [115]. Due to its success in many domains, it is promising to also evaluate multi-task learning for LUR models, making use of colocated air pollution data [41].

A number of publications have already considered multi-task learning for air pollution modeling. Xu and Yoneda [155] conduct air pollution modeling with multi-task learning. However, their multiple tasks use the same pollutants but at different stations, and they consider an air pollution forecasting task. Chen et al. [28] propose an air pollution model that uses multi-task learning in that it simultaneously tries to spatially estimate and forecast air quality index levels. In contrast to this thesis' contribution, it does not

Table 2.3.: Multi-task learning for LUR results in Zurich. Average R^2-scores on the test samples from the OpenSense dataset using MLPs with different numbers of shared layers. The increase is calculated between the single-task learning model (zero shared layers) and the best performing multi-task learning model (at least one shared layer). Performance of comparison LUR models is also provided for reference. Best MLP values are written in bold. [41]

	MLP shared layers				Comparison methods	
Samples	0	1	2	Increase	Linear Regression	RF
100	0.224	0.169	**0.224**	+0.41%	0.131	0.262
300	0.410	**0.448**	0.391	+9.23%	0.250	0.475
500	0.463	**0.474**	0.379	+2.26%	0.264	0.566

make use of colocated air pollutant data, considers time-series data instead of annual mean data, and it solves a classification instead of a regression task (air quality index levels are discrete classes). Li et al. [82] does employ multi-task learning where each task considers a different pollutant, but also in a time-series forecasting setting instead of LUR. However, they also find improvements when making use of colocated pollution data for their time-series task.

2.1.2.2. Multi-Task Learning for LUR

In order to make use of colocated sensors and alleviate label sparsity, the multi-task learning approach is evaluated for MLPs in LUR, meaning that models are trained to estimate multiple different pollutants at the same time. Experiments are conducted with a MLP consisting of two hidden layers and varying number of hidden layers shared between all tasks, starting from a model where all hidden layers are shared among the tasks, over a model where one hidden layer is shared and another hidden layer is specific to each task, and finally a single-task model, where all hidden layers are task-specific. The corresponding paper "Evaluating the multi-task learning approach for land use regression modelling of air pollution" (see Appendix A.2, [41]) was published in 2020 in the Journal of Physics: Conference Series and presented at the second International Conference on Frontiers of Artificial Intelligence and Machine Learning (FAIML), where it received the "Best Student Paper" award.

Multi-task learning for LUR is evaluated based on the OpenSense dataset containing calibrated and filtered UFP measurements and factory pre-calibrated CO and O_3 measurements from a mobile measuring campaign in Zurich that attached sensors on top of tram cars [57]. The CO and O_3 measurements were filtered as well as possible for this work, but there was no accurate reference data that would be needed for calibration, leading to lower data quality. Furthermore, modeled NO_2, NOx, and particulate matter (PM) concentrations for Central London from the LAEI [6] are used which do not exhibit data quality issues. The LUR models are trained to estimate annual mean air pollutant concentrations. For evaluation, ten-fold cross-validation is conducted and mean metrics

Table 2.4.: Multi-task learning for LUR results in London. Average R^2-scores on the test samples from the London dataset using MLPs with different numbers of shared layers. The increase is calculated between the single-task learning model (zero shared layers) and the best performing multi-task learning model (at least one shared layer). Performance of comparison LUR models is also provided for reference. Best MLP values are written in bold. [41]

	MLP shared layers				Comparison methods	
Samples	0	1	2	Increase	Linear Regression	RF
100	0.489	**0.490**	0.476	+0.32%	0.459	0.477
300	**0.506**	0.468	0.490	−3.09%	0.488	0.527
500	0.514	**0.515**	0.507	+0.18%	0.499	0.537
3000	0.522	0.528	**0.534**	+2.25%	0.505	0.572

across the folds are reported. Results show that multi-task learning can improve model performance significantly when sufficient data is available. On the data from Zurich, Table 2.3 shows that multi-task learning can improve performance by up to 9.23 %, while Table 2.4 shows more mixed results but still mostly improvements on the London data. Furthermore, the MLPs consistently outperform simple linear regression baselines and are close to RF models, but multi-task learning is not enough to let MLPs catch up to the RF performance. Since performance gains only show with relatively large datasets, multi-task learning seems not to be suited for lowering data requirements for more complex LUR models, but it may improve already working models when measurements for different pollutants are available.

2.1.2.3. Recent Related Work

To the best of my knowledge, no other work before or after this publication considered making use of colocated sensors for training LUR models. Multi-task learning has also not been applied in this LUR setting in any other way in the works known to me. Thus, this contribution presents a novel approach to LUR that is useful to consider when conducting future LUR studies.

2.2. Climate

In order to produce more accurate climate data, DL can be an important tool if new methods for this domain are developed. Two research questions asked in this thesis consider DL for climate modeling and this work's contributions aim to answer these questions. The following contextualizes these contributions within related work with regard to their research question.

2.2.1. Can Deep Learning Models Improve the Quality of Climate Model Outputs?

Climate models are not perfect, which can affect the quality of their outputs [92]. Modern DL models can improve these outputs in a data-driven way. The following describes prior related work on improving climate model outputs with ML and DL approaches. Next, two contributions of this thesis are presented which aim to improve climate model outputs with DL models, namely ConvMOS [133, 134] and semi-supervised learning for grain size distribution interpolation [75]. Finally, recent related work regarding the research question and the presented contributions is addressed.

2.2.1.1. Prior Related Work

A well-established task where DL is used to improve the quality of climate data is *perfect-prognosis statistical downscaling* [12]. This task aims to estimate local, often high-resolution climate data from relatively coarse climate data. The term "perfect-prognosis" stems from the fact that these models use so-called *reanalysis data*, which is not the direct output of a climate model but instead data that combines past observations and model data in order to obtain "perfect" predictors. Thus, these models aim to learn the relationship between coarse, large-scale atmospheric variables and local variables. Such models can improve climate model outputs by increasing their spatial resolution, but errors made by a climate model are unlikely to be corrected since these models are trained with "perfect" data and not with a climate model's imperfect outputs. It is common to use CNNs for this task [12, 107, 145].

This thesis considers two general approaches to improving climate model outputs where DL methods have been less common: post-processing a climate model's outputs through climate model output statistics (MOS) and, more indirectly, by generating better input data, namely soil data, that can be used by climate models using spatial interpolation techniques. The following addresses prior related work to these two approaches.

Climate MOS The first approach — climate MOS — is related to the aforementioned perfect-prognosis statistical downscaling task, but MOS considers the outputs of a concrete climate model as an input instead of "perfect" data. This allows a climate MOS method to learn specific error characteristics of the climate model in order to reduce these errors in climate model outputs. Climate MOS has been worked on before in either a distribution-wise or an event-wise manner. For the former type, distribution characteristics like means or variances of the climate model's output are mapped to the observed distribution [43]. The latter type directly maps the individual simulated events (e.g., specific precipitation simulations for each day) to the observed events (e.g., the observed precipitation amounts for each day) with statistical and ML models, which tends to perform better than distribution-wise climate MOS [43]. The following focuses on event-wise MOS for its improved performance. A rather simple approach that has shown to work reasonably well is local linear regression. With this method a linear regression is fitted for each location of interest to estimate the observed precipitation per

timeframe (e.g., per day) based on the simulated precipitation per timeframe at that location [43]. More common are non-local MOS approaches where one also fits one model per location, but that model is also provided with data from locations in the vicinity. Many conventional ML methods have been used for non-local MOS in recent years.

One such technqiue commonly used is linear regression. It is typically combined with principal component analysis (PCA), where it is then often called principal component regression (PCR). PCA reduces the dimensionality of the input features and the resulting leading principal components are fed into the linear regression model. While this PCA-based pre-processing step has its own name when combined with linear regression, it is also frequently applied when using other conventional ML models for climate MOS. An example for the successful application of PCR for climate MOS is Eden and Widmann [43] who applied it to the general circulation model (GCM) ECHAM5. Paeth [106] also used PCR to post-process precipitation of the regional climate model (RCM) REMO (which is also the climate model used in the experiments in this thesis) [43, 106].

Simple NNs have also been used for climate MOS. For example, Moghim and Bras [96] proposed the use of a three-layer MLP, which they evaluated using the GCM CCSM3.

Precipitation of GCMs has been corrected with RF models by Sa'adi et al. [125] and Noor et al. [102].

Support vector machines (SVMs) were also used for climate MOS with GCMs. Precipitation was corrected with this model in Bangladesh by Pour et al. [110], in Borneo by Sa'adi et al. [125], and in Pakistan by Ahmed et al. [5], for example.

Ahmed et al. [5]'s approach does not only use SVMs, but it combines these with RFs. The former correct the output of a GCM and the latter combine the outputs of multiple SVMs, with each considering an own GCM, to form a sort of multimodel ensemble of data corrected with MOS. However, RFs have also been used successfully for climate MOS by themselves by Noor et al. [102], for example. Sa'adi et al. [125] not only evaluated SVMs, but also RFs, finding that both models provide decent accuracy with SVMs still providing a lower error.

In order to make use of the performance potential of DL techniques and the special properties of the climate MOS task as a geospatial environmental regression task, the ConvMOS architecture was developed (see Appendices A.3 and A.4, [133, 134]) [133].

Spatial Interpolation For the second approach, one particular climate model input that can benefit from ML is soil data. To this end, grain size distribution interpolation is conducted, providing grain size distributions in the soil for all locations of interest in a study area based on relatively few soil samples that are available for some locations. These distributions affect hydrological processes and are therefore important inputs for climate models. Typical approaches to environmental geospatial interpolation tasks like grain size distribution interpolation are techniques like k-nearest neighbor (kNN), inverse distance weighting (IDW), or Kriging [83]. However, these methods can usually not directly incorporate additional information available at the locations of interest and many are not able to model non-linearities. To alleviate these issues, prior work has successfully considered NNs for interpolation which can include other predictors and

model non-linearities [35, 119, 139]. One issue these NN models have is their relatively large demand for labeled data points [54], which are only sparsely available for grain size distribution data. In this context, this thesis presents a novel interpolation method based on NNs with the label sparsity issue in mind [75], where I have contributed to in a limited manner. Such improved techniques provide more accurate soil data which may then improve the quality of climate models using this data.

2.2.1.2. ConvMOS

ConvMOS is a novel convolutional DL architecture designed specifically with the nature of typical errors of climate models in mind. These errors are often either location-specific due to poor grid point representation of land surface characteristics like topography [106] or systematic due to simplified climate processes, so-called parameterizations [106]. ConvMOS targets both error types efficiently by combining per-location model parameters for location-specific errors and global model parameters for systematic geospatial error patterns based on CNN layers. To this end, a global module based on 2D convolutional layers is proposed and a local module, that uses a 1D convolution to effectively train per-location parameters. ConvMOS models then consist of a sequential combination of such modules. This architecture was first proposed in the paper "Deep Learning for Climate Model Output Statistics" (see Appendix A.3, [134]) which was presented at the "Tackling Climate Change with Machine Learning" workshop at NeurIPS 2020 where it won the "Best ML Innovation" award. More refinements on the architecture, additional experiments and a more in-depth analysis are proposed in the paper "ConvMOS: Climate Model Output Statistics with Deep Learning" (see Appendix A.4, [133]) which was published in the Springer journal Data Mining and Knowledge Discovery as part of the special issue of the journal track for the conference ECML/PKDD. This latter work also considers the popular U-Net architecture [123] as a global module, which is then called CM-UNet (ConvMOS U-Net).

To evaluate ConvMOS, experiments are conducted on the daily outputs of the RCM REMO [66, 67, 91] for the period from 2000 to 2015 in a study area that spans over an extended German region with 0.11° resolution (−1.43° to 22.22° E and 42.77° to 57.06° N). The model aims to produce precipitation estimates that fit as close as possible to the observational dataset E-OBS [58] version 19.0e, which is a gridded dataset based on an ensemble of interpolated station data and also contains data for the extended German region for the period from 2000 to 2015. The dataset is split into a training (years 2000-2009), a validation (year 2010), and a test set (years 2011-2015) [133, 134].

An architecture composition study evaluates different combinations of the proposed global and local modules. Crucially, it shows that models which have both types of modules — and thus combine per-location and global model parameters — are able to provide better performance than models that only consist of global or local modules [133].

The MOS approaches with non-deterministic fitting procedures (i.e., all except models based on linear regression) were trained and tested 20 times and the mean metrics with standard deviations are reported. Table 2.5 compares ConvMOS models (ConvMOS and CM-UNet) and other baselines including the conventional climate MOS approaches local

Table 2.5.: ConvMOS results. Mean metrics on the test set for all study area locations having observational data. Values rounded to two decimal places. Std. dev. for correlation (always 0.00) and skill score (between 0.00 and 0.03) omitted for brevity. RMSE and NRMSE represent the average squared error, with the former expressed in mm of precipitation and the latter expressed in percent of the range of possible precipitation values at a location. Pearson correlation (Cor.) measures the linear correlation between observed and estimated precipitation, with a value of one being best and a value of zero indicating no linear correlation. Correlation mean is calculated with Fisher's z-transformation [131]. The skill score considers the precipitation distributions and measures the similarity between the estimated and observed probability distribution function, with a value of one being best and zero being worst. R^2 describes how much of the target variable's variance is explained by the model with a value of one being perfect, a value of zero being the performance achieved by always estimating the mean, and lower values imply even worse performance. Bias is the mean difference between model estimates and labels, which is ideally zero. A positive bias indicates that the model tends to produce larger values than the labels, while a negative bias implies the opposite. Best values are written in bold. [133]

MOS \ Metric	RMSE [mm]	NRMSE [%]	Cor.	Skill	R^2	Bias [mm]
REMO raw	5.32	15.83	0.49	**0.91**	−28.24	0.31
Lin	3.51	8.03	0.58	0.47	0.33	−0.03
NL PCR	3.37	7.80	0.62	0.81	0.36	0.02
NL RF	3.39 ± 0.00	7.82 ± 0.00	0.61	0.82	0.36 ± 0.00	0.03 ± 0.00
ResNet18	3.03 ± 0.01	7.04 ± 0.03	0.71	0.60	0.47 ± 0.01	-0.06 ± 0.07
ResNet34	3.06 ± 0.02	7.10 ± 0.04	0.71	0.61	0.46 ± 0.01	-0.07 ± 0.09
ResNet50	3.04 ± 0.01	7.05 ± 0.03	0.71	0.61	0.47 ± 0.00	-0.10 ± 0.10
ResNet101	3.03 ± 0.02	7.04 ± 0.04	0.71	0.64	0.47 ± 0.01	-0.04 ± 0.08
U-Net	2.97 ± 0.02	8.37 ± 0.12	**0.74**	0.82	-5.60 ± 0.88	-0.03 ± 0.08
CM-UNet	$\mathbf{2.92 \pm 0.01}$	7.01 ± 0.11	**0.74**	0.70	0.13 ± 0.22	$\mathbf{0.01 \pm 0.10}$
ConvMOS	2.93 ± 0.02	$\mathbf{6.77 \pm 0.05}$	0.73	0.89	$\mathbf{0.51 \pm 0.02}$	-0.10 ± 0.05

linear regression (Lin), non-local PCR (NL PCR), and non-local RF (NL RF), but also popular standard CNN architectures like ResNets [60] and U-Net [123] that have not been developed for climate MOS in particular. The comparison is done using a number of metrics, including the square root of the mean squared error called RMSE, a normalized RMSE expressed in percentage of each location's range of possible precipitation values, Pearson correlation, the skill score proposed by Perkins et al. [109], R^2 which measures the portion of the target's variation that is explained by the model (one is perfect and zero is the result when always estimating the test data's mean), and Bias which is simply the mean error, indicating tendencies towards over- or underestimation. The skill score calculates the common area between the probability density function of the observed and estimated precipitation. As such, it considers the precipitation distributions' similarity where one would be a perfect score [109]. The results show that both ConvMOS models which combine local and global model weights tend to perform best, with regular ConvMOS providing considerably better NRMSE, skill score, and R^2 while CM-UNet provides slightly better RMSE, Correlation, and Bias. All MOS approaches improve each metric except for the skill score, which is reduced by all MOS methods, suggesting that REMO's precipitation distribution at land locations is already rather close to that of the observations with a skill score of 0.93 and can barely be improved by MOS methods [133].

2.2.1.3. Semi-Supervised Learning for Grain Size Distribution Interpolation

The quality of climate model outputs can not only be improved by directly reducing errors in the outputs, like described in the previous section on ConvMOS (see Section 2.2.1.2). Providing a climate model with more accurate input data may also improve it. One such input that some climate models make use of is grain size distribution data for soil which can be used to model soil-hydrological processes. In order to estimate more accurate soil data, this thesis presents an approach that considers semi-supervised learning as a strategy to cope with label sparsity common in environmental ML tasks, which means that a model is not only trained with labeled samples but also using samples that have no label. The proposed method first makes use of the unlabeled samples by generating weak labels for each location with a conventional interpolation technique with stronger modeling assumptions, then trains a MLP on these weak labels, and finally fine-tunes the MLP on the actual labeled dataset. Despite not all locations having labels, auxiliary data (e.g., altitude, long-term means for precipitation and temperature) is available everywhere, providing the model with additional information which it can make use of. The corresponding paper "Semi-Supervised Learning for Grain Size Distribution Interpolation" (see Appendix B.2, [75]) was presented at the "Machine Learning Advances Environmental Science" (MAES) workshop at the International Conference on Pattern Recognition (ICPR) 2020 and published by Springer in the ICPR conference proceedings in 2021. Note that in contrast to the papers concerning ConvMOS, I have contributed to this work only as co-author and therefore provided only limited contributions [75].

For experimental evaluation, a soil profile database from the Bavarian Environment Agency is used that is not publically available [14]. It contains grain size distributions for 315 locations in Lower Franconia with soil information from a depth of 14 cm to 15 cm,

Table 2.6.: Test results (mean ± standard deviation) for each interpolation model. The metrics are mean absolute error (MAE), mean squared error (MSE), and Jensen-Shannon divergence (JSD). "PT" stands for pre-training. "FT" stands for fine-tuning. Best values are written in bold. [75]

Model	MAE	MSE	JSD
Mean	0.5210 ± 0.0384	0.1337 ± 0.0183	$0.054\,90 \pm 0.007\,63$
kNN	0.4267 ± 0.0412	0.1011 ± 0.0223	$0.039\,81 \pm 0.009\,03$
IDW	0.4188 ± 0.0417	0.0954 ± 0.0225	$0.038\,15 \pm 0.009\,05$
MLP	0.4361 ± 0.0552	0.1068 ± 0.0251	$0.042\,56 \pm 0.008\,83$
SemiMLP (after PT)	0.4781 ± 0.0577	0.1296 ± 0.0283	$0.049\,74 \pm 0.009\,95$
SemiMLP (after FT)	$\mathbf{0.4078 \pm 0.0445}$	$\mathbf{0.0952 \pm 0.0195}$	$\mathbf{0.037\,72 \pm 0.007\,65}$

which is the depth for which the model is supposed to estimate grain size distributions. Additionally, meteorological data from the German Meteorological Service [38] as well as topographic information from the Bavarian Environment Agency [13] are used to provide the model with additional information about both labeled and unlabeled locations. All methods are evaluated using a ten-fold cross-validation. The results in Table 2.6 show that the semi-supervised MLP (SemiMLP) provides the best performance in comparison to the baseline interpolation methods IDW and kNN. IDW is the best baseline interpolation technique but a Wilcoxon signed-rank test [150] with a significance level of 0.01 indicates that the fine-tuned SemiMLP outperforms it significantly. The combination of pre-training on data with weak labels and fine-tuning on the labeled samples performs better than plain supervised training (see row "MLP"), showing that it is an effective training strategy in this setting. An analysis also finds considerable performance drops when permuting the values of some features like precipitation and temperature, suggesting that these features are important to the model and thus help the model's estimation quality [75].

2.2.1.4. Recent Related Work

Climate MOS Recently, a number of other works were published for climate MOS or closely related fields that also proposed novel approaches and models.

Shortly after this thesis' contribution "Deep Learning for Climate Model Output Statistics" (see Appendix A.3, [134]) was published, Grönquist et al. [55] proposed post-processing ensemble weather forecasts instead of climate data with CNNs and locally connected networks, that has similarities to the ConvMOS architecture. They used the U-Net [123] architecture and found good results, which is why we also considered a similar model for climate MOS in the follow-up paper "ConvMOS: Climate Model Output Statistics with Deep Learning" (see Appendix A.4, [133]).

François et al. [50] proposed post-processing of climate model outputs with cycle-consistent adversarial networks based on CNNs. They compared their model to several baseline methods not based on ML and found favorable results, but they did not compare

their method to event-wise ML MOS approaches considered in this thesis.

Watt-Meyer et al. [149] recently proposed an interesting use-case for MOS in which they trained a RF to correct simulations towards observations and then coupled this model to a weather model, correcting variables at each timestep. They found no instabilities in the model and reduced errors in precipitation outputs.

Spatial Interpolation No other recent work considered grain size distribution interpolation specifically, but numerous novel model approaches were proposed for spatial interpolation with other geospatial data lately, which may also be used to interpolate more accurate input data for climate models.

Zhu et al. [162] propose the novel conditional encoder-decoder generative adversarial network (CEDGAN) architecture which combines an encoder-decoder structure with adversarial learning and generative models for spatial interpolation. They compare their method to the conventional interpolation techniques IDW and ordinary kriging on elevation interpolation in China, finding superior performance with their method. The authors aim to improve spatial interpolation with a more complex model architecture while this thesis' contribution, in contrast, proposes a different training strategy for a comparatively simple MLP model.

Otake et al. [105] consider DL models based on long short-term memory (LSTM) cells for seismic intensity data. They feed the model with data from nearby permanently installed seismometers and use a temporary seismometer at a location of interest to train the model to estimate seismic intensity there. While they use the term spatial interpolation for their work, it is not a classical interpolation technique in that it does need at least temporary measurements for locations of interest in contrast to other methods.

Dauji and Rafi [37] use a MLP for the spatial interpolation of data from standard penetration tests (SPTs). The MLP is trained to estimate the SPT value at a location of interest based on the distance to the nearest neighbors and their respective SPT values. They compared their model to conventional interpolation techniques, finding improved performance with their MLP. While their model is structurally similar to the model used in this thesis' contribution (both are MLPs) they do not explore semi-supervised learning to further improve model performance.

Shi and Wang [128] consider ensembles of NNs with radial basis activation functions for spatial interpolation, which allows their model to quantify uncertainty. They find better estimation of spatial patterns and prediction uncertainty given enough data in comparison to baseline approaches, but they provide no comparison with other techniques based on NNs.

Sekulić et al. [127] propose a spatial interpolation technique based on RFs called Random Forest Spatial Interpolation. They find that it mostly outperforms baseline methods, but they have not considered interpolation techniques based on NNs.

Another work that considers a RF variant for spatial interpolation is Maxwell et al. [93]. They use quantile regression forests to spatially model coal properties. These models can quantify uncertainty and show better performance than IDW and similar performance

to regression kriging in their experiment on data from a coal mine in Australia. They have also not considered techniques using NNs for comparison.

Li [86] proposes the enhanced dual IDW method based on IDW. It is in particular suitable for data exhibiting clusters by integrating data-to-data correlation with conventional IDW and is further enhanced by incorporating locally varying exponents instead of fixed global exponents that alter the effective distance of existing samples to an unseen location of interest. Li [86] finds improved performance compared to IDW and dual IDW without locally varying exponents as well as more robust estimates compared to ordinary kriging. Again, no comparisons to interpolation techniques based on NNs were conducted.

2.2.2. Can Deep Learning Models be Taught to Better Estimate Climate Extremes?

When using DL models for the estimation of climate variables like precipitation, they are likely to have issues in estimating extremes, since their typical training procedures effectively reward them for providing conservative estimates closer to the mean [133]. Climate extremes can have severe consequences for life on earth [42], which is why it is of interest to improve DL models' estimation of climate extremes. The following presents prior work related to DL for climate extreme estimation and regression with imbalanced target distributions. Then, density-based weighting for imbalanced regression [135] and its application to the ConvMOS model [133] are presented, which show how to teach DL models to better estimate climate extremes. Thereafter, recent related work regarding the research question and the contributions are discussed.

2.2.2.1. Prior Related Work

In related work, there are some DL approaches that consider climate extremes, but they aim not to estimate climate extremes themselves well but forecast weather-patterns that are associated with extremes instead, like El Niño/Southern Oscillation (ENSO) events [56]. Chattopadhyay et al. [26] developed an analog forecasting system with a DL model for extreme-causing weather patterns. This can avoid the aforementioned issue since these patterns may not exhibit extreme climate variable values themselves, but it may limit the model to detecting specific extreme events (e.g., ENSO) instead of extremes in general. Other work uses DL models to detect extremes like tropical cyclones in climate model outputs [73], but they do not estimate climate variables themselves.

The core issue to solve for the good estimation of climate extremes with ML is data imbalance. Data imbalance is an issue for many ML algorithms, since these models tend to become biased towards focusing more on samples with commonly occurring labels than samples with rare labels [76]. In the literature, there are two general approaches to alleviate this issue [76]: Data-level methods over- and/or undersample a dataset partly to achieve a more balanced distribution. Algorithm-level methods adapt learning algorithms to deal with the issue. A wealth of publications propose techniques to cope with data imbalance for classification tasks. Notable data-level methods include ADASYN [59] and

SMOTE [27], which propose strategies to create new samples for rare classes and remove samples of common classes, leading to more balanced data. Algorithm-level methods are usually based on cost-sensitive learning and often weight samples according to the inverse class frequency [63, 148]. Up until recently, there have been very few publications that proposed similar methods for regression tasks, all of which were only data-level methods, namely SMOTE for regression (SmoteR) [140] and SMOGN [17]. In this context, this thesis' contributions of the density-based sample weighting method for imbalanced regression DenseWeight and, building upon DenseWeight, the cost-sensitive learning approach for imbalanced regression DenseLoss were developed (see Appendix A.5, [135]). This technique is used in the second publication regarding the convolutional DL climate MOS model ConvMOS (see Appendix A.4, [133]) in order to improve estimation quality for climate extremes.

2.2.2.2. Density-based Weighting for Imbalanced Regression

DenseWeight is a sample weighting approach for imbalanced regression which first estimates the density function of the training target values $p(y)$ through kernel density estimation (KDE). This density function is normalized to lie between zero and one with min-max normalization, providing the normalized density function $p'(y)$. Then, weights are calculated for each of the N training data points based on the corresponding target value's normalized density with DenseWeight's weighting function

$$f_{\mathrm{w}}(\alpha, y) = \frac{\max(1 - \alpha p'(y), \epsilon)}{\frac{1}{N}\sum_{i=1}^{N}(\max(1 - \alpha p'(y_i), \epsilon))} \,. \tag{2.1}$$

The hyperparameter ϵ defines the lower bound for the possible weights, that is set so 10^{-6} in the experiments to avoid zero and negative weights. The weighting function is scaled by the hyperparameter α, which influences the severity of weighting differences between common and rare samples ($\alpha = 0$ disables density-based weighting, larger α increases weighting differences between rare and common samples). Intuitively, this weighting scheme leads to samples in rarer parts of the target variable range receiving larger weights than those in more common parts. DenseLoss is a cost-sensitive learning method for imbalanced regression which assigns these weights to the samples by weighting a loss function during the model training of a NN. A more thorough explanation can be found in the corresponding paper "Density-based weighting for imbalanced regression" (see Appendix A.5, [135]) which was published in 2021 in the Springer journal Machine Learning as part of the special issue of the journal track for the conference ECML/PKDD 2021, where the work has also been presented at [135].

In this work, three main experiments were conducted: one experiment on synthetic data to confirm that DenseLoss works as expected, another experiment on benchmark datasets to compare it to the best approach to imbalanced regression previously available SMOGN [17], and an experiment with an existing CNN-based DL model for statistical downscaling of precipitation called DeepSD [145], to show that DenseLoss works on an environmental real-world task with a DL model. An additional experiment was conducted where SMOGN is adapted to use DenseWeight as its relevance function in

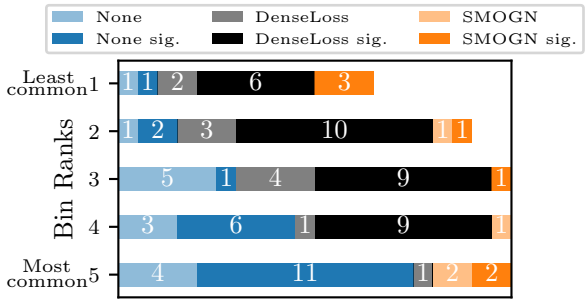


Figure 2.1.: Number of datasets from the 20 comparison datasets for which each imbalanced regression method (DenseLoss, SMOGN, and None) leads to the best performance per bin based on RMSE. Bins are ranked within each of the 20 test datasets according to the number of data points. Bins with rank 1 (5) contain the fewest (most) samples. Each bar section shows the number of datasets for which a particular method provides the best performance (i.e., "wins") at that bin rank. When a method's "wins" are denoted as "sig." they are significant regarding both other methods. Five test datasets had a bin without data points and two test datasets had two bins without samples. Because of this the bars for bin rank 1 and 2 are smaller as no winner can be determined for empty bins. DenseLoss leads to the best performance most often in comparison to the sampling-based method SMOGN and applying no imbalanced regression method (None) for the four rarest bin ranks (bin ranks 1 to 4). [135]

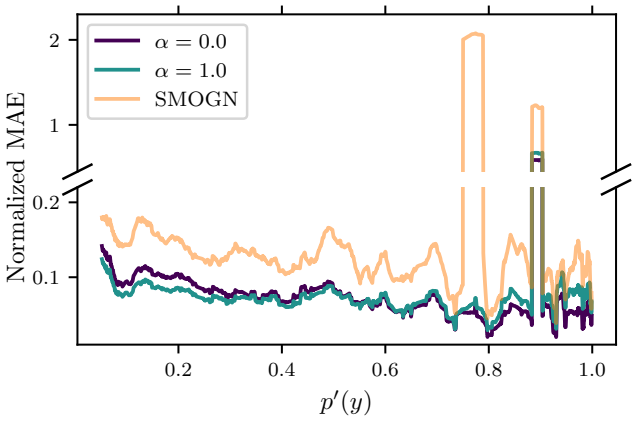


Figure 2.2.: Normalized MAE (i.e., MAE divided by the difference between maximum and minimum target value of a particular sample's dataset) for test samples from the 20 comparison datasets per normalized density when applying no method for imbalanced regression ($\alpha = 0.0$), DenseLoss ($\alpha = 1.0$), or the sampling-based method SMOGN. The graph is smoothed via moving mean (window size 300) for interpretability. Samples with higher normalized density have more common target values than those with lower normalized density, which are rarer. DenseLoss leads to lower errors for relatively rare samples ($\sim p'(y) < 0.5$) while increasing error for common samples, shifting the model's focus towards rare samples. SMOGN does not provide good performance over the 20 datasets regardless of sample rarity. Note that the two outlier segments stem each from a single sample of the *availPwr* dataset that have a very high feature value, leading to very large estimates and high MAE. [135]

order to find out whether DenseLoss's superior performance stems from the different relevance functions or from the difference between resampling and reweighting. Each dataset used is split into a training, a validation, and a test set. Each model considered in an experiment is trained and tested 20 times per dataset and mean metrics are reported. For the first two experiments and the final additional experiment, a simple MLP with three hidden layers and ten neurons each with ReLU activation [101] is used as the main model, while additional MLP structures are also evaluated in the paper's Appendix, which show similar results as the main architecture. The last experiment uses the CNN-based DL model for statistical downscaling of precipitation DeepSD [145] and mostly adheres to its original experimental setup [135].

For the first experiment, synthetic datasets with different imbalanced distributions are generated, showing that DenseLoss works as expected, emphasizing performance of rare samples to a greater extent with increasing α [135].

In the second experiment, DenseLoss is compared to the best approach to imbalanced regression previously available SMOGN, using the same 20 benchmark datasets evaluated in the original SMOGN paper [17]. For evaluation, each dataset's test data points are binned based on their target value, with each bin spanning 20 % of the target value range in the test set. These bins are then ranked within each dataset by the number of samples, so that the bin with the lowest number of samples has bin rank 1 and the bin with the highest number of samples has bin rank 5. In this way, it is possible to compare performance of similarly rare samples across datasets by considering performance metrics per bin rank. Figure 2.1 shows the number of datasets of the 20 comparison datasets for which each imbalanced regression method leads to the best performance (i.e., "wins") per bin rank. "Wins" denoted as "sig." are statistically significant regarding both other methods according to a Wilcoxon signed-rank test [150] and a significance level of 0.05. Bin ranks 1 and 2 do not sum up to 20, since some bins were empty and, accordingly, no winner could be determined there. DenseLoss most frequently provides the best performance for the four rarest bin ranks (bin ranks 1 to 4), outperforming the sampling-based method SMOGN. As expected, applying no method for imbalanced regression typically performs best for the most common bins with rank 5.

To also consider the different imbalanced regression methods' absolute performance over the varying density of the 20 datasets, a continuous analysis can be used. Figure 2.2 plots each test data point's mean absolute error (MAE) — min-max normalized to lie between zero and one based on the particular dataset's minimum and maximum target values — with regard to its normalized density $p'(y)$ within its dataset. Thus, the plot visualizes the MAE across all 20 datasets depending on the data point rarity, which is expressed in the normalized density $p'(y)$, for training without an imbalanced regression technique ($\alpha = 0.0$), DenseLoss ($\alpha = 1.0$), and the sampling-based method SMOGN. As with the binned evaluation, DenseLoss again improves model performance for relatively rare datapoints ($\sim p'(y) < 0.5$) while it is best to apply no imbalanced regression method for optimal performance of more common samples. Thus, DenseLoss can shift a model's focus towards rare samples. SMOGN is not able to perform well on average across the 20 datasets, regardless of sample rarity [135].

The final main experiment with DenseLoss and the CNN-based DL model for statistical

downscaling of precipitation DeepSD [145] also finds performance gains for rare samples in addition to unusual performance improvements for very common samples. This shows the applicability of the proposed method to environmental ML tasks with relatively large DL models and large datasets, to which SMOGN could not be applied as it is computationally infeasible for the size of the dataset [135].

The experiments show superior performance with DenseLoss in comparison to SMOGN, but it is also interesting to investigate where this difference comes from. To this end, SMOGN is adapted to use DenseWeight as its relevance function. An additional experiment on the 20 comparison datasets shows that the performance difference between SMOGN and DenseLoss seems to be mostly due to the difference between resampling and reweighting and not due to different relevance functions, because SMOGN with DenseWeight as its relevance function was not able to reach or improve upon DenseLoss's performance [135].

To apply density-based weighting to training models better suited for climate extremes, the ConvMOS model is considered. In the second publication on this matter (see Appendix A.4, [133]), analysis shows that the DL approaches considerably underestimate the number of high precipitation events. Thus, an experiment is conducted using ConvMOS with the cost-sensitive learning technique for imbalanced regression DenseLoss, in order to shift the focus on better performance for rare extreme samples. For evaluation, the same setting as described in Section 2.2.1.2 for ConvMOS is used, except that the test samples are binned into a lower precipitation bin (less than 50 mm precipitation) and a high precipitation bin (at least 50 mm precipitation), while ConvMOS is still trained to solve a regression task and output concrete precipitation values. The ConvMOS model with DenseLoss improves RMSE in the high precipitation bin while performing worse in the lower precipitation bin and it also "classifies" the samples better in the two bins by having a balanced accuracy of 60.49 % instead of regular ConvMOS' balanced accuracy of 55.97 %. This is also slightly better than the RCM REMO's balanced accuracy for these two bins of 59.90 % [133].

2.2.2.3. Recent Related Work

While I am not aware of other work considering the improved estimation of climate extremes with DL models, a number of other researchers took interest in imbalanced regression and published additional work related to this topic.

Shortly prior to the publication of the cost-sensitive learning technique for imbalanced regression DenseLoss, Ribeiro and Moniz [120] proposed the evaluation metric SERA that can assess the effectiveness of models for the prediction of extreme values in imbalanced regression and may also be used to optimize models for extreme value prediction. In contrast to this thesis' contribution, their method does not use the density function to provide a measure for data point rarity but instead relies on an adjusted box-plot method. They also do not use their technique to directly influence model training via cost-sensitive learning.

Almost concurrently to the publication of "Density-based weighting for imbalanced regression", Yang et al. [157] proposed label distribution smoothing (LDS) and feature

distribution smoothing (FDS) to cope with data imbalance for regression tasks. LDS also estimates the target variable's density through KDE, and they use the resulting density function for cost-sensitive reweighting through simple multiplication with the inverse density and a square-root variant thereof. DenseWeight, in contrast, proposes a more complex function than the inverse that is specifically tailored with gradient-based learning methods in mind. FDS bins the target variable and smooths the feature distributions between nearby bins, so that a continuity in target space corresponds to continuity in feature space, which is an interesting approach less related to this thesis' contribution. The authors find mostly superior performance in comparison to the sampling approaches SmoteR [140] and SMOGN [17], but they have not compared their methods to the approaches proposed in this thesis.

Thereafter, Ren et al. [118] proposed the balanced mean squared error (MSE) for imbalanced regression, specifically with computer vision tasks like age estimation or pose estimation in mind. The balanced MSE does not directly calculate sample weights before the optimization like with DenseWeight or LDS, but it instead includes a balancing term that aims to restore balanced predictions. The authors compared their method not to DenseWeight or DenseLoss but to LDS from Yang et al. [157] and find that it outperforms LDS.

To the best of my knowledge, none of these works considered applying their methods to the improved estimation of climate extremes. Thus, it may be interesting future work to consider these techniques as well.

3. Conclusion

The design of novel machine learning (ML) and especially deep learning (DL) approaches for environmental domains was considered in this thesis. The following provides a summary and gives an outlook for potential future work in this area.

3.1. Summary

This thesis presented ways of designing novel approaches for environmental ML tasks especially based on DL. To this end, common properties of typical environmental ML tasks have been identified, namely geospatiality, continuous variables, data imbalance, colocated sensors, and spatial label sparsity. These properties have to be kept in mind when developing novel ML approaches in this domain since they lead to certain challenges but also opportunities.

In contrast to most existing approaches for these tasks, which employ conventional ML techniques, this thesis specifically looked to contribute novel DL approaches for their benefits, such as their potential for exceptional performance and their automatic feature engineering capabilities. To ideally cope with the challenges and opportunities typically presented by environmental data when developing ML and DL methods, this thesis suggests the use of (i) convolutions to exploit the geospatiality of environmental data and efficiently learn spatial patterns, (ii) regression models to estimate the typically continuous environmental variables, (iii) density-based weighting for better estimation quality for rare and extreme samples with imbalanced data when these samples are of particular interest, (iv) multi-task learning to profit from the availability of multiple related target variables due to the often colocated sensors, and (v) semi-supervised learning to alleviate issues related to label sparsity.

Using these techniques and DL, this thesis considers four research questions which demonstrate how to develop powerful, novel ML approaches in the environmental domain: (i) *Can air pollution be estimated without manual feature engineering?* The land use regression (LUR) model MapLUR based on convolutional neural networks (CNNs) as well as the off-the-shelf LUR solution OpenLUR showed two ways of accomplishing this. (ii) *Can colocated pollution data improve spatial air pollution models?* The proposed multi-task learning approach for LUR showed potential improvements. (iii) *Can DL models improve the quality of climate model outputs?* ConvMOS, the DL climate model output statistics (MOS) architecture presented in this thesis, directly improved the quality of climate model outputs. Additionally, the semi-supervised training of multilayer perceptrons (MLPs) with weak labels for grain size distribution interpolation was able to provide improved input data for climate models. (iv) *Can DL models be taught to better estimate climate extremes?* This thesis first proposes density-based weighting

for imbalanced regression (DenseLoss) which is then used in conjunction with the DL climate MOS architecture ConvMOS to estimate climate extremes more accurately.

These research questions and the contributions that answered them, helped to further improve estimation quality for their specific tasks and also showed new ways of approaching the tasks, which may open paths for future work to advance even further. Such advances in environmental ML tasks provide us with more accurate information on the environment's state. Based on this information, we can better understand the environmental issues we are experiencing and plan steps towards mitigation and adaptation more effectively.

3.2. Outlook

While this thesis has presented a multitude of novel approaches to environmental ML tasks, there are still many tasks where in particular DL techniques have potential to improve estimation quality or maybe even allow for completely new approaches. Rolnick et al. [122] provide an overview of tasks related to climate change where ML is promising, many of which also are environmental ML tasks as described in this thesis, for example, tasks related to peatland monitoring, modeling ice sheet dynamics and sea level rise, or urban building energy models.

One aspect of this thesis was air pollution modelling and in particular LUR. There are LUR models of different complexity which tend to provide better performance the more complex they are. However, with increased complexity the data requirements also tend to grow, which reduces the applicability. Thus, an interesting direction for future work is to develop techniques that lower data requirements of more complex LUR models. Another interesting LUR approach for future work may be the combination of dispersion models and LUR as done in some initial prior work [142].

Another aspect this thesis regarded in particular were environmental ML tasks related to climate modeling. For future work, there may of course still be room for improvement in post-processing climate data with MOS. ML and DL in particular have very active research communities continuously developing novel methods that may be even more suitable for environmental ML tasks with their typical properties in the future. Another interesting approach, other than post-processing, is learning better parameterizations than those currently in use by climate models, which may lead to the climate model directly producing more accurate forecasts. As of writing this, the latter approach could be demonstrated to work to some extent in relatively simple settings but not yet in complex climate models that are used in practice [116, 158]. The same is true for the even more challenging approach of learning entire climate models with neural networks (NNs) which is also not feasible yet [40]. In order to provide climate models with better base data for their forecasts related to the hydrological cycles, grain size distribution interpolation may also be advanced further in the future, for example, by modeling the depth dimension. This would allow for more complete 3D soil information that would also be useful for climate models and may also further enhance estimation quality [75].

A. Main Publications

MapLUR: Exploring a new Paradigm for Estimating Air Pollution using Deep Learning on Map Images

MICHAEL STEININGER, University of Würzburg, Germany
KONSTANTIN KOBS, University of Würzburg, Germany
ALBIN ZEHE, University of Würzburg, Germany
FLORIAN LAUTENSCHLAGER, University of Würzburg, Germany
MARTIN BECKER, Stanford University, USA
ANDREAS HOTHO, University of Würzburg, Germany

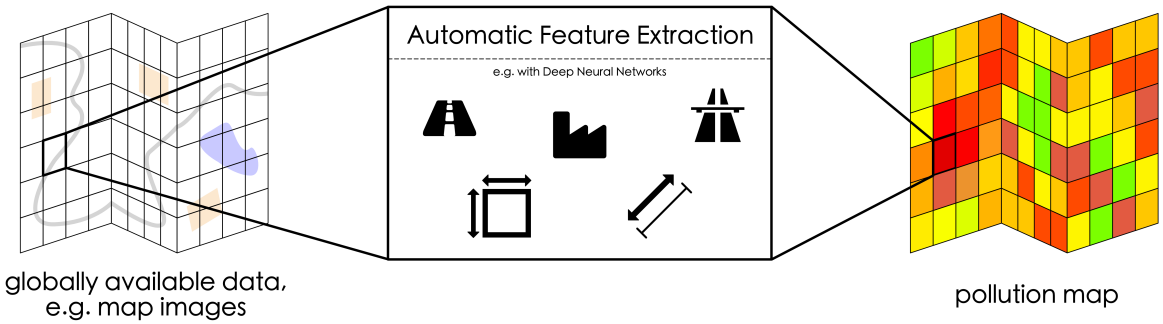


Fig. 1. **MapLUR: Automatic feature extraction and globally available data for air pollution modeling.** We propose the DOG paradigm (Data-driven, Open, Global) for land-use regression which advocates openly and globally available data, and automatically extracting features in order to estimate pollution. Following this newly introduced paradigm, we propose the MapLUR model. It consists of a deep neural network architecture that estimates pollution concentrations for specific locations directly from globally available map images (rendered maps or satellite images) resulting in area spanning pollution maps. For this, MapLUR automatically learns to extract features from the given map images. The extracted features closely resemble manually engineered features for land-use regression models.

Land-use regression (LUR) models are important for the assessment of air pollution concentrations in areas without measurement stations. While many such models exist, they often use manually constructed features

Authors' addresses: Michael Steininger, University of Würzburg, Institute of Computer Science, Chair of Computer Science X, Würzburg, Germany, steininger@informatik.uni-wuerzburg.de; Konstantin Kobs, University of Würzburg, Institute of Computer Science, Chair of Computer Science X, Würzburg, Germany, kobs@informatik.uni-wuerzburg.de; Albin Zehe, University of Würzburg, Institute of Computer Science, Chair of Computer Science X, Würzburg, Germany, zehe@informatik.uni-wuerzburg.de; Florian Lautenschlager, University of Würzburg, Institute of Computer Science, Chair of Computer Science X, Würzburg, Germany, lautenschlager@informatik.uni-wuerzburg.de; Martin Becker, Stanford University, Stanford Medicine, Nima Aghaeepour Lab, Stanford, USA, mgbckr@stanford.edu; Andreas Hotho, University of Würzburg, Institute of Computer Science, Chair of Computer Science X, Würzburg, Germany, hotho@informatik.uni-wuerzburg.de.

2374-0353/2020/1-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

based on restricted, locally available data. Thus, they are typically hard to reproduce and challenging to adapt to areas beyond those they have been developed for.

In this paper, we advocate a paradigm shift for LUR models: We propose the **D**ata-driven, **O**pen, **G**lobal (DOG) paradigm that entails models based on purely data-driven approaches using only openly and globally available data. Progress within this paradigm will alleviate the need for experts to adapt models to the local characteristics of the available data sources and thus facilitate the generalizability of air pollution models to new areas on a global scale.

In order to illustrate the feasibility of the DOG paradigm for LUR, we introduce a deep learning model called MapLUR. It is based on a convolutional neural network architecture and is trained exclusively on globally and openly available map data without requiring manual feature engineering. We compare our model to state-of-the-art baselines like linear regression, random forests and multi-layer perceptrons using a large data set of modeled NO_2 concentrations in Central London. Our results show that MapLUR significantly outperforms these approaches even though they are provided with manually tailored features.

Furthermore, we illustrate that the automatic feature extraction inherent to models based on the DOG paradigm can learn features that are readily interpretable and closely resemble those commonly used in traditional LUR approaches.

CCS Concepts: • **Computing methodologies** → **Neural networks**; *Image representations*; Supervised learning by regression; • **Applied computing** → **Environmental sciences**.

Additional Key Words and Phrases: land-use regression, air pollution, deep learning

ACM Reference Format:
Michael Steininger, Konstantin Kobs, Albin Zehe, Florian Lautenschlager, Martin Becker, and Andreas Hotho. 2020. MapLUR: Exploring a new Paradigm for Estimating Air Pollution using Deep Learning on Map Images. *ACM Trans. Spatial Algorithms Syst.* 1, 1 (January 2020), 24 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Air pollution is known to have adverse effects on human health and the environment [11, 22]. Thus, especially in areas with high population counts, it is important to control local pollution concentrations. For this reason, monitoring stations are deployed in many cities, which measure pollution continuously in order to assess whether the pollution levels are still within acceptable/legal limits. However, since the number of stations in a city is usually very limited, there are many areas where no air quality data is available. To fill these gaps, land-use regression (LUR) models are often used to estimate pollution concentrations in areas without monitoring stations [7, 26, 54].

Problem Setting. In recent years, a wealth of different land-use regression models have been developed that have shown to provide promising pollution estimates. However these models i) are at least partially based on neither globally nor openly available data [3, 7, 27] and ii) often rely on hand-crafted features.

Thus, due to the local nature of the features, i) these models usually do not generalize easily to locations other than the one they were developed for. Additionally, due to the involved hand-crafting process, ii) optimizing the features for new models in specific study areas is a cumbersome process.

Approach. To address the challenges inherent to inaccessible data and manual feature engineering in land-use regression models, in this work, we advocate a paradigm shift towards purely data-driven land-use regression models based on open and globally available data. We call the corresponding paradigm **DOG** (**D**ata-driven, **O**pen, **G**lobal). More specifically, models adhering to DOG work directly on raw data, automatically extracting their features from the input. While such data-driven methods have proven successful in multiple application domains [33], they have so far not been introduced to land-use regression. Land-use regression models following the DOG paradigm

have multiple advantages: i) they can be fit more easily to different study areas than other, more specialized land-use regression approaches, ii) they do not require manual feature engineering, and iii) they can be reproduced by other researchers without requiring access to data sources that are not easily available. In order to demonstrate the feasibility of this paradigm, we introduce the MapLUR model. MapLUR implements DOG by using deep learning, specifically a convolutional neural network architecture. It automatically extracts features from map images, which are openly available almost anywhere on earth, and estimates air pollution based on these features.

Experimental Evaluation. We assess the performance of MapLUR by comparing it against state-of-the-art land-use regression models like linear regression, Random Forests (RF), and Multi-layer Perceptrons (MLP) on modeled NO_2 concentration data from the London Atmospheric Emissions Inventory (LAEI) [2]. In the process, we employ different types of images including map images from OpenStreetMap and Google Maps [24] as well as satellite imagery from Google Maps. We find that our model works best using map images from OpenStreetMap and that it outperforms all baselines significantly.

Furthermore, we analyze the data requirements of MapLUR and the baselines. We find that common for deep learning models, MapLUR requires more training data than models that rely on hand-crafted features. In this context, we evaluate how far the training set can be reduced and discuss possible approaches to further address this challenge.

Finally, we analyze the automatically extracted features by observing which parts of the map images were particularly important for our model using guided backpropagation [51] and artificial map images. The analysis shows that the learned MapLUR features strongly relate to hand-crafted features as commonly used in land-use regression models.

Contribution. Our core contributions in this work are threefold:

(1) We propose DOG, a new, data-driven paradigm to land-use regression. Models following this paradigm should not require manual feature engineering and only rely on openly and globally available data sources.
(2) We introduce MapLUR, a land-use-regression model based on DOG. MapLUR employs a deep learning approach to automatically extract features from map images. We show that this model is able to outperform traditional land-use regression models when trained on a sufficiently large data set.
(3) We demonstrate that, contrary to popular believe, models based on the data-driven paradigm are not necessarily black-boxes by inspecting the features MapLUR extracts, finding that the automatically extracted features strongly relate to typical manually engineered features for land-use regression models.

Structure. This work is organized as follows. Related work is summarized in Section 2. Section 3 describes the air pollution data and image data used in this work. DOG and the MapLUR model are introduced in Section 4. The experiments and the baseline models are described in Section 5. Section 6 presents our results and analyzes our model. We discuss advantages and limitations of DOG and MapLUR in Section 7. Finally, Section 8 concludes this work.

2 RELATED WORK

Land-use regression has been an active field of research for many years now. Work done in the previous decade has laid important foundations for current land-use regression models and established linear regression techniques as the de facto standard model [7, 43, 54]. Especially noteworthy is the Escape project [7, 20] which built models for 36 European areas. The model building procedure of this project has become a standard approach [40, 41, 43, 52, 53]. In order

to make the application of land-use regression models easier, there is a tool available which automizes the process of variable generation, modeling and prediction with a model based on linear regression [42].

However, more advanced machine learning methods are starting to become more common. One example for these approaches are Random Forests [9]. They have been used successfully to estimate elemental components of particulate matter in Cincinnati, Ohio [10] and NO_2 pollution in Geneva [14].

Another example for a more advanced method are neural networks. These models have been used to estimate a range of pollutants successfully, as shown in various publications. For example, they have been applied to NO_2 [1, 14, 37], $PM_{2.5}$ [1, 5, 6, 23, 55], PM_{10} [3, 37, 55], and surface dust [12] concentrations. The neural-network-based models are typically simple multi-layer Perceptrons. However, there are deep learning models which use recurrent neural networks or deep belief regression networks. These models differ from this work in that they are used to forecast pollution concentrations from earlier measurements or fill missing values for locations where measurements already exist [5, 6, 23, 55], while we estimate pollution for locations without measurements. To the best of our knowledge, there are no deep learning models for our setting. Both Random Forests and neural networks have been shown to outperform linear regression in land-use regression [10, 14].

Support vector regression models [19] are another possible approach. There are models which can forecast pollution concentrations using this technique [35, 47], but there do not seem to be any land-use regression models with this type of model.

All aforementioned land-use regression models rely on manually engineered features, which are typically gathered from various locally available data sources that might not be available elsewhere. In contrast to all methods above, we propose a deep learning model based on convolutional neural networks (CNNs), which is able to automatically learn relevant features from openly available maps.

Such image-based approaches have been used before in the context of air quality estimation and pollution detection. Singh [50] interpreted modeled air pollution data as images and used non-machine-learning image classification techniques in order to detect higher pollution episodes. Furthermore, CNNs have been used before in the context of air quality estimation by Zhang et al. [57] and Li et al. [36], who proposed models to estimate air haze level using photos from, for example, mobile phones or webcams. In contrast, our work uses map and satellite imagery depicting land-use as model input, making our model more closely related to land-use regression models. Additionally, our model estimates pollution concentrations instead of haze levels.

3 MATERIALS

In this section, we introduce the air pollution data set we use to train and evaluate our method as well as the data sources from which we extract map and satellite images.

3.1 Air Pollution Data

We train and test our model using pollutant concentrations from the London Atmospheric Emissions Inventory (LAEI) [2]. It contains modeled annual mean concentrations of NO_2 and PM_{10}, among other pollutants, at a 20 m grid level for the complete Greater London area in 2013. For our main model development and evaluation we use the NO_2 concentrations of the data set since it is a very frequently used pollutant for land-use regression models. The data is the result of a dispersion model which incorporates a vast number of input factors like for example road and rail networks, traffic data, aviation, pollution from individual industrial premises, domestic and commercial fuel consumption, as well as fires. Through this approach, 5,856,428 data points were generated where each data point represents a 20 m by 20 m cell [2].

Table 1. **Statistics of London's NO_2 concentrations.** The statistics for Central London include all LAEI cells in the bounding box. The sampled data set contains randomly sampled cells from the Central London data set, which are used for training and testing models in our experiments. Mean, standard deviation (SD), minimum values (Min), and maximum values (Max) are in $\mu g/m^3$.

Data set	Count	Mean	SD	Min	Max
Central London	113,680	50.90	15.02	37.12	253.89
Sampled subset	4,500	50.85	15.02	37.17	171.06

We sample a training set consisting of 3,000 data points and a test set consisting of 1,500 data points from the Central London part of the data set in order to have a reasonable number of urban data points for our experiments. We choose data points from Central London because we believe that it is more important in practice to reliably estimate pollutant concentrations in highly polluted areas with a large population than in more rural areas. For this, we define a geographical rectangle that roughly describes Central London and only use cells within. The box's north western corner is at 526,660 easting and 183,220 northing while the south eastern corner is at 534,760 easting and 177,640 northing specified in British National Grid coordinates. The sampled data points are depicted in Figure 8 in Appendix A. Descriptive statistics for the cells in Central London and the sampled subset used for training and testing can be found in Table 1.

The map images, which we use to depict the areas of the data points, show 80 m by 80 m areas even though the air pollution data is available at a 20 m grid level. The 20 m by 20 m cells are in the center of these images. This allows MapLUR to see more of the surroundings and incorporate information about distant emission sources. In order to avoid a potential evaluation issue, we sample data points in such a way that no images can overlap. Any overlap could lead to a situation where the model already roughly knows the pollution concentration for a test data point since it might recognize the test data point's area from the image of a nearby training data point. Such implicitly learned proximity of data points could give our model an unfair advantage, which we avoid with our procedure.

3.2 Image Data

This section describes the sources of map and satellite image data that we use in this paper as well as the preprocessing applied to the images in order to generate training samples.

3.2.1 Image Sources. There is a variety of globally available sources for map images, two popular services being Google Maps [24] and OpenStreetMap [46]. While Google Maps is a commercial and proprietary service, OpenStreetMap is an open database for map data that is built and maintained by volunteers. Data from OpenStreetMap can be used to render maps in various ways through different stylesheets. In this work, we render map images based on OpenStreetMap data using a slightly modified version of the default stylesheets used on the official OpenStreetMap website. It differs from the default in that we do not render text like street or station names, since labels obstruct map features making them harder to recognize and often only carry very localized information, thus, possibly reducing generalizability. For tile rendering we use *mod_tile* which is a module for the Apache web server with the rendering back-end *renderd* [45]. In addition to OpenStreetMap images, we use map and satellite images from Google Maps, in order to compare the effectiveness of each visualization for this task. Since Google Maps is proprietary, the images cannot be easily customized to the same extent as OpenStreetMap images. We therefore use them without modification.

3.2.2 Image Preparation. Before using MapLUR it is necessary to prepare map or satellite images. We found through preliminary experiments that images depicting 80 m by 80 m provide the best performance in this setting, as can be seen in Appendix B. To depict the correct area in an image we approximate the 80 m distances using a meter per pixel value that depends on the zoom level of the image and the latitude of the location due to the Mercator projection, which both OpenStreetMap and Google Maps use. We obtain the images at zoom level 17, resulting in an pixel extent of approximately 0.75 m by 0.75 m at London's latitude. Thus, the images have a resolution of 106 px by 106 px. The rendered images are then scaled to a fixed resolution of 224 px by 224 px similar to what popular CNN architectures for the ImageNet competition [18] use. This way we avoid having to change the model architecture when depicting different sized areas in the images or when we fit the model to locations with a different latitude which would also result in differently sized images. Using a model input resolution that fits the images exactly could reduce the model size and improve training and inference speed. Nonetheless, we found it unnecessary considering the already acceptable speed and we favored the increased flexibility.

4 METHODS

In the following, we introduce our data-driven paradigm DOG to land-use regression, which suggests that models should estimate air pollution by automatically extracting relevant features from openly and globally available data. We also present our model *MapLUR*, which follows this paradigm by using a convolutional neural network as an automatic feature extractor, taking as input globally and openly available map images.

4.1 The DOG Paradigm

Our data-driven paradigm DOG (Data-driven, Open, Global) aims to alleviate the issues of manual feature engineering as well as only locally applicable models. To this end, it requires models to fulfill the following criteria:

- **Automatic extraction of relevant features**: Models should learn a function from the raw input to the desired output, which leads to automatic development of features and may even uncover relevant factors that are not yet known as having an influence on air pollution. This can for example be achieved by deep learning methods like the convolutional neural network introduced in the next section.
- **Usage of globally available data sources**: Models should rely exclusively on data sources that are available for (almost) all parts of the world. This allows the ubiquitous application of the model without collecting additional, only locally available data sources.
- **Usage of openly available data sources**: Models should rely exclusively on openly available data sources. This allows researchers to reproduce and improve the model's results without requiring access to paid or not publicly available resources.

4.2 The MapLUR Model

In this section, we propose the specific model MapLUR based on the DOG paradigm described above. Our model applies the paradigm by using map images as a globally, openly available source of information and extracting features through deep learning, more specifically a convolutional neural network (CNN). This type of network is a natural fit for our setting, since it is specifically designed to work with two-dimensional shapes like images. CNNs utilize spatial locality within the images and reduce the number of learned parameters through weight sharing. These concepts make it feasible to learn relevant features from raw images, in contrast to fully-connected networks, which would need an impractical amount of parameters [34].

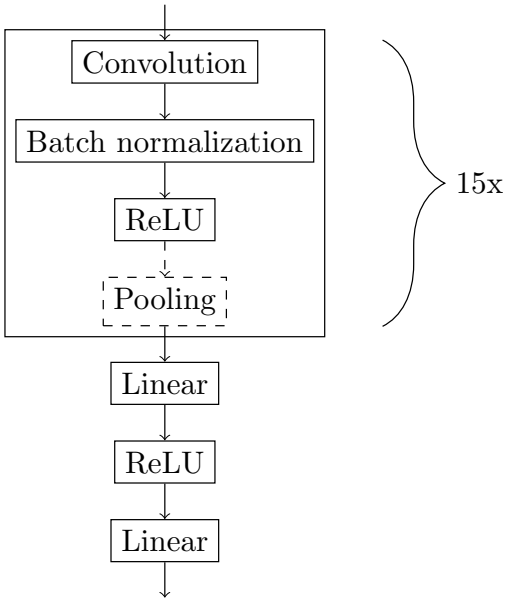


Fig. 2. **Structure of MapLUR.** The model consists of 15 feature-learning building blocks which contain a convolution layer, batch normalization, rectified linear units (ReLUs), and sometimes a pooling layer. These building blocks are concatenated and only the first, third, fifth, seventh, tenth, and thirteenth block contain a pooling layer. These blocks are followed by a simple fully connected layer with ReLU activation and finally a single fully connected neuron with linear activation which returns the estimation of the pollution at the given location.

The structure of MapLUR is depicted in Figure 2. It contains 15 convolutional layers with batch normalization [29] and rectified linear units (ReLUs), which are pair-wise linear activations [44]. The last convolutional layer is followed by a fully connected layer with 128 neurons and ReLU activation (depicted as the third and second to last boxes in Figure 2). These neurons are then connected to a single neuron with linear activation that produces the estimated pollution concentration. Each convolutional layer has 16 filters, a kernel size of 3, a padding of 1, a dilation of 1, and a stride of 1. The output size of these layers is the same as their input size. Maximum pooling layers with a kernel size of 2 and stride of 2 are applied after the ReLUs of the first, third, fifth, seventh, tenth, and thirteenth convolutional layer in order to reduce the number of activations. We found this architecture and the corresponding hyperparameters by evaluating different variations of the model using ten fold cross-validations on the training set.

We use ReLU activations since they have shown to work well for many different tasks and models, making them the most popular activation function for deep learning applications [33]. While trying different architectures we have also experimented with SELU [31] and RReLU [56] activations but we found no improvements with these functions. The linear activation in the final layer is common practice for regression tasks [32]. It does not restrict the range of resulting values allowing the model to estimate any value.

5 EVALUATION

In order to evaluate MapLUR, we conduct several experiments and compare our model to baseline models which are commonly used in land-use regression. The experiments and the baseline models are described in the following.

5.1 Experimental Setting

We conduct four experiments using MapLUR, varying the data available to the model. For all experiments, MapLUR is trained using the Adam optimizer [30] on batches of size 400 for at most 2,000 epochs with a learning rate of 0.0001. We augment the training data by flipping or transposing the images. Additionally, we employ early stopping, interrupting training when the validation performance has not increased for 20 epochs in a row.

Experiment 1 — OpenStreetMap. In the first experiment, only OpenStreetMap images are used as input to the CNN. The input images have three channels (RGB), 224 px by 224 px, and depict an area of 80 m by 80 m. All labels were removed from the rendering process of the images, as described in Section 3.2.1.

Experiment 2 — Google Maps. Instead of OpenStreetMap images, Google Maps images were captured and fed into the model in this experiment. The same size as in the previous experiment was used. As Google Maps data is proprietary, modifications cannot be made as easily and to the same extent as with OpenStreetMap images. Therefore, text labels are present in the imagery.

Experiment 3 — Google Maps Satellite. For the third experiment, instead of stylized map images, we use satellite images from Google Maps Satellite [24] which uses imagery from both satellites and aerial surveys. Training and test images from Google Maps Satellite have the same size and zoom levels as the OpenStreetMap and the Google Maps images.

Experiment 4 — OpenStreetMap and Google Maps Satellite. Experiment 4 then combines OpenStreetMap images and satellite images by concatenating the two RGB images to one six-channel tensor.

5.1.1 Evaluation Setup. The models are evaluated using standard metrics for the evaluation of land-use regression models, namely R^2 and RMSE. Both metrics are explained in Appendix C. In all experiments, the model is initialized and trained 40 times on the training set and evaluated on the test set, both of which are described in Section 3.1. The average of the resulting evaluation metrics is then used as the final score to counteract unfortunate initialization results. Additionally, the sample of 40 evaluation runs can be used as the input to statistical significance tests to formally confirm differences in evaluation results.

5.2 Baselines

In order to determine how well our model works, it is necessary to compare it to other methods. Therefore, we first describe a set of features that is used by our baseline models. Thereafter, four baseline models are introduced, namely a mean baseline, linear regression, Random Forest, and multi-layer Perceptron. The last three of the aforementioned models are commonly used in land-use regression. Random Forests and multi-layer Perceptrons tend to yield state-of-the-art results as described in Section 2.

5.2.1 Features. We use a set of standard land-use and road-related features for our baseline models. These features have shown to be important influencing factors for air pollution [20]. All of these features can be calculated from OpenStreetMap data, since we want to provide similar information to all models for a fair comparison. The features include the areas of commercial, industrial, and residential land-use, the lengths of big and local streets, and the distances to the next traffic signal, motorway, primary road, and industrial premise. Big streets include streets that are classified as either motorway, trunk road, primary road or secondary road in OpenStreetMap while all other streets are local streets. Most features are typically calculated for different *buffers*, which are areas

with a specific radius around data points. For example, the areas of different types of land-use and the lengths of streets are calculated for 50 m and 100 m buffers in order to give the baseline models similar sight into the surroundings as the MapLUR model. However, the features which calculate the distance from each data point to specific locations like the closest traffic signals, roads or industrial premises exist only once and are not calculated for different buffers. Due to these features, the baseline models are given a slight advantage since they can get information from entities which are further away than MapLUR can see.

5.2.2 Mean. A simple baseline for a regression task is the mean baseline. It disregards all features and estimates the mean value of all training data points for each test data point. This baseline provides performance values that every other model should beat.

5.2.3 Linear regression. The most common approach to land-use regression is linear regression. Therefore, it is useful to compare our novel model to this type of model.

We use the same supervised stepwise selection as Eeftens et al. [20] for selecting the most relevant subset of features. A description of this procedure can be found in Appendix D.1. After applying the stepwise selection on the development set the model is left with the variables *length of big streets* (50 m buffer), *distance to the next industrial premise*, and *distance to the next traffic signal*.

5.2.4 Random Forest. The Random Forest is a more powerful model that was shown to work well for land-use regression and can often provide better performance than typical linear regression approaches, as it can model non-linear correlations between features [10, 14]. Therefore, we use it as another baseline in this work.

This model is built in a similar way to the procedure in Brokamp et al. [10]. Details are in Appendix D.2. The final Random Forest model uses the variables *distance to the next industrial premise*, *distance to the next primary road*, *distance to the next traffic signal*, *distance to the next motorway*, *length of big streets* (50 m buffer), and *area of residential land-use* (100 m buffer). It builds 394 trees using bootstrap samples, considers at most 42.79 % of the available features per split, needs at least three samples to split a node, and needs at least three samples for a leaf node.

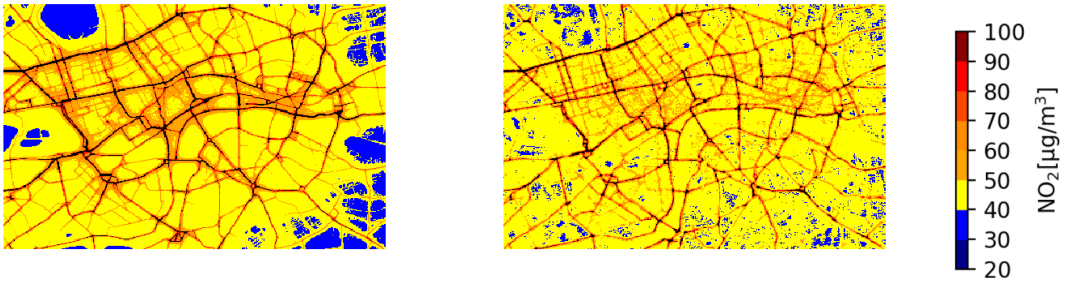
5.2.5 Multi-layer Perceptron. Neural networks, or more specifically multi-layer Perceptrons (MLPs), are models whose popularity for land-use regression tasks has grown in recent years and which often outperform other baselines [1, 3, 12, 14, 37]. Additionally, evaluating multi-layer Perceptrons (even if not directly applied to image data) illustrates the performance of neural networks which, in contrast to the MapLUR model, are not based on convolutions.

Again, we follow the model development procedure of previously published work. In this case, we base our procedure on the one used by Alam and McNabola [3]. Appendix D.3 contains a description of this procedure.

The MLPs use the variables *distance to the next industrial premise*, *distance to the next primary road*, *distance to the next traffic signal*, *distance to the next motorway*, *length of big streets* (50 m buffer), *length of local streets* (100 m buffer), *area of industrial land-use* (100 m buffer), *area of commercial land-use* (100 m buffer), and *area of residential land-use* (100 m buffer). The architecture search found that the best performing MLP model has a single hidden layer with 29 neurons. This is similar to the MLPs used by previous publications [1, 3, 12, 14, 37].

6 RESULTS AND ANALYSIS

Given the baseline methods and MapLUR's description, we now present the results for our experiments and analyze MapLUR in terms of data requirements and features learned.



(a) LAEI data (b) MapLUR estimates

Fig. 3. **Comparison of LAEI data and MapLUR estimates.** The estimated map shares strong similarities with the original data. It can be seen that MapLUR is able to recognize streets and accurately associate them with high pollution concentrations. However, our model tends to overestimate pollution in areas with very low pollution concentrations and underestimate pollution for areas with no road close by.

6.1 Experiments

Table 2 shows the results of the baseline methods as well as MapLUR's results for our experiments. All results in the Table are significantly different from each other. To verify this, the metrics of each model are tested for normality using the test from D'Agostino and Pearson [15, 16] with $p < 0.05$. The statistical significance for models whose metrics are normally distributed are tested using a t-test, while the other models' metrics are tested with the Wilcoxon signed-rank test, both testing for $p < 0.05$. Additionally, Bonferroni correction [8] is applied, which further substantiates the statistical significance, since $p < \frac{0.05}{n}$ with $n = 7$ for each model pair. $n = 7$ is chosen to account for the number of hypotheses that are tested on the same data (each model is tested against 7 other models).

Baselines. As described before, we use a simple mean baseline, a linear regression, an approach with Random Forests, and a multi-layer Perceptron with manually engineered features from OpenStreetMap data. The results in Table 2 show that the Random Forest is performing considerably and significantly better than both the linear regression and the MLP.

Experiment 1 — OpenStreetMap. Our model with OpenStreetMap images performs better than all baselines regardless of metric, which can be seen in Table 2.

Figure 3 shows the original NO_2 concentrations of the LAEI data set in Central London and the estimates of MapLUR. It can be seen that our model is able to come rather close to the original data using only OpenStreetMap images, but tends to overestimate values of areas with low pollution concentrations and underestimate values of areas which are not in the vicinity of roads.

Experiment 2 — Google Maps. This experiment uses Google Maps imagery instead of OpenStreetMap images. A drop in R^2 of more than 10 percentage points in comparison to the previous experiment and a higher RMSE value may be explained by the styling of Google Maps images. Googles Maps contain fewer color-coded entities. Especially streets, that are a common entity for land-use regression features, are not diversified as much as in OpenStreetMap images. The differences can be seen in Appendix E.

Experiment 3 — Google Maps Satellite. This experiment uses Google Maps Satellite imagery as input. Table 2 shows that using only Google Maps Satellite imagery leads to a considerable drop in

Table 2. **Results of baseline methods and experiments.** MapLUR with OpenStreetMap images is providing the best performance overall, beating all baselines and all other MapLUR variants. Using satellite images from Google Maps instead of OpenStreetMap images decreases the metric scores on the evaluation set. Combining both image types does not improve the score. The Random Forest model is outperforming all other baseline models on this data set which makes it the best baseline. All results are significantly different to each other.

Model	R^2	RMSE [$\mu g/m^3$]
Mean baseline	0.000	13.971
Linear regression	0.487	10.004
Multi-layer Perceptron	0.499	9.887
Random Forest	0.662	8.119
MapLUR experiment 1: OpenStreetMap	**0.673**	**8.002**
MapLUR experiment 2: Google Maps	0.537	8.918
MapLUR experiment 3: Google Maps Satellite	0.206	12.389
MapLUR experiment 4: OpenStreetMap and Google Maps Satellite	0.660	8.112

performance, even worse than the linear regression baseline with an R^2 of 0.206 and an RMSE of 12.389.

These results are most likely due to the noise in the satellite images, which makes it harder to discern influencing factors for air pollution. The hand-labeled map images therefore help a lot as they already encode the desired entity labels as colors.

Experiment 4 — OpenStreetMap and Google Maps Satellite. The last experiment combines map and satellite imagery by concatenating both three-channel RGB images to one six-channel tensor. OpenStreetMap images are used for the map images, since they have shown better performance than Google Maps in our task. Satellite imagery is taken from Google Maps Satellite. As both image sources use the same spatial resolution of 80 m by 80 m, local OpenStreetMap data should only be augmented by the satellite images. However, the results in Table 2 show no performance gain compared to using OpenStreetMap only. In fact, the results are worse and significantly different for both models.

Computation times. The computation times for the various models evaluated vary due to the different model complexities. For example, with an Intel Xeon E5-2690V4 CPU training the MLP takes on average 25 seconds, Random Forest trains on average for 11 seconds, and the linear regression model is typically built in a single second. Estimating pollution concentrations with these simple trained models for an area like Central London only takes seconds. MapLUR is the most complex model among them, but can be trained in reasonable time on commodity hardware. We found that training it on a single consumer graphics card (Nvidia GTX 1080 TI) takes about 50 minutes. Once MapLUR is trained it can estimate pollution concentrations for the complete Central London area in 35 seconds.

6.2 Model Analysis

After seeing that MapLUR can work well, we now further analyze the model. First, we assess the data requirements of MapLUR. Then, we demonstrate that our model can be made interpretable

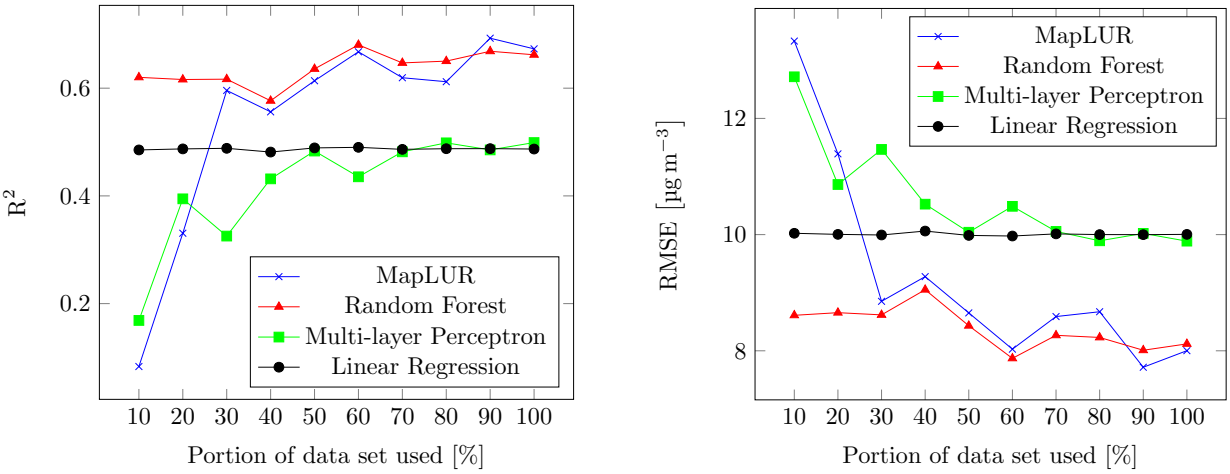


Fig. 4. **Data requirements of land-use regression models.** These graphs show the performance of commonly used land-use regression models and MapLUR with varying training data set sizes. The portion of the data set used refers to the size of the NO_2 data set with which the models were trained in our experiments. Thus, 100 % is equivalent to 3,000 data points. Each point in the graphs is the mean of 40 model runs, except for MapLUR's points which we only ran 5 times per data set size due to the model's computational complexity. This shows that MapLUR can provide comparable results with 900 data points and it tends to improve with more data. Multi-layer Perceptrons behave similarly but they need more data to reach other baselines. Linear Regression and Random Forests are less dependent on data set size.

by analyzing what our model has learned through guided backpropagation and by creating fake OpenStreetMap images.

6.2.1 Analyzing Data Requirements. The previous experiments showed that our model can successfully model air pollution in a data-driven way. While this is the main focus of this paper, the data we use (3,000 data points) is larger than those typically available in a real world setting. In the following, we analyze the actual data requirements of all models evaluated above. Overall, the corresponding results will inform future studies on data requirements and point towards necessary methodological advancements.

When gradually reducing the number of training data points, we noticed a drop in performance with smaller data sets for all models except for linear regression (cf. Figure 4). Models based on neural networks experience a more pronounced performance loss in comparison to, for example, Random Forests, where there is only a slight decline. We believe that this stems from the size and complexity of these models compared to simpler models like linear regression models. More parameters need to be trained which tends to require more training examples. However, about 30 % of the training data is still sufficient for our model to exhibit performance comparable to the strongest baseline, which is the Random Forest trained on hand-crafted features.

Addressing the increased need for data is an important point for future work, which we also discuss in some more detail in Section 7.

6.2.2 Understanding Estimates using Guided Backpropagation. In this section, we apply a technique called *guided backpropagation* [51], which allows us to visualize the regions of the input image that the network focuses on for its estimation. This approach starts off with a forward pass of an image. Thereafter, the gradient of the activation is computed with respect to the input image. At each ReLU in the model, positive gradients whose corresponding output during the forward pass was

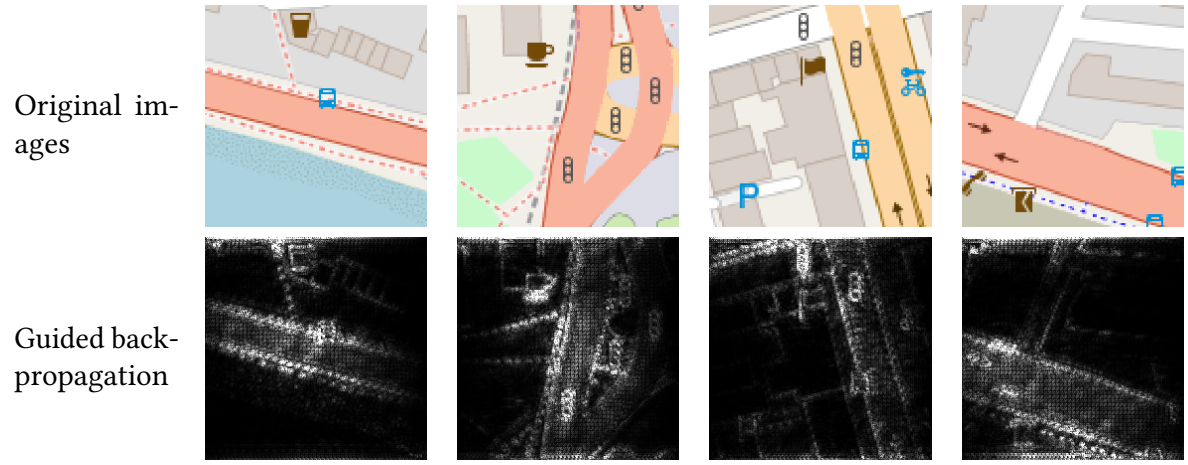


Fig. 5. **Visualization of detected features using guided backpropagation [51].** This technique highlights important pixels in the input images by visualizing gradients of the activation with respect to pixel intensities. All negative gradients and positive gradients, whose corresponding output during the forward pass was negative, are set to 0 at each ReLU during backpropagation. This approach reveals parts of an image which contribute to the pollution. As one would expect, the model is concentrating on large streets. (Original images: © OpenStreetMap contributors)

negative and negative gradients are set to 0 so that only features which contribute to the estimated pollution concentration are shown. This allows us to visualize which parts of the image the model is paying attention to. Several examples are shown in Figure 5.

The guided backpropagation shows that the model is paying special attention to motorways, trunk roads, and primary roads which are rendered in red or orange colors in OpenStreetMap. This shows that MapLUR is able to automatically learn intuitively relevant features, since traffic is known as a large factor for NO_2 pollution [13]. MapLUR also considers buildings, foot paths and cycle paths to some extent for its estimates while it seems to be ignoring water and park areas. The model tends to pay more attention to pixels close to the center, which is understandable since we estimate the pollution concentration for the 20 m by 20 m areas that are in the center of each image.

6.2.3 Analyzing Entity Influence using Artificial Map Tiles. One of the biggest advantages of using a DOG-based model for air pollution estimation is that it extracts features by itself, while previous work always used hand-engineered features. The leading question in developing land-use regression methods in previous work is: *What entity* of *what area* in *what distance* to the center is contributing to the pollution? From this, three categories of features arise: entity features, area features, and distance features. We now want to investigate the correlation of these features with the model's output. For this, we take advantage of the well-defined structure of map images with different color-coded entities. Map images therefore can easily be recreated using graphic editing software, which makes it possible to create artificial OpenStreetMap images for which we can control the features separately while keeping all other features fixed. We then observe changes to the model's output while modifying the values of these features.

Entity Features: Entity features describe *what* is seen on the image. Entity features are often used for the estimation of air pollution, as, for example, industrial areas are usually contributing more to air pollution values than parks. In this experiment, we investigate how certain entities are influencing the model estimate. We build two kinds of images: On the one hand, we create images that are each completely covered by one specific type of entity, resulting in uniformly

Table 3. **Model estimate for a given OpenStreetMap entity.** The entities span across the whole image and they are overlaid with different types of roads. Overlaying a road with another road does not make sense so these values are omitted. The 'neutral' entity is a background that is used by OpenStreetMap for indicating land with no particular land-use. All estimates are in $\mu g/m^3$.

Road type	Entity Name								
	industrial area	residential area	commercial area	park	forest	water	neutral	motorway	trunk
no road	37.71	38.29	38.72	38.87	39.27	41.66	42.06	47.23	80.63
trunk	61.70	50.94	59.73	64.14	57.94	59.62	58.62	—	—
motorway	60.04	48.48	64.18	46.05	54.21	53.45	55.00	—	—

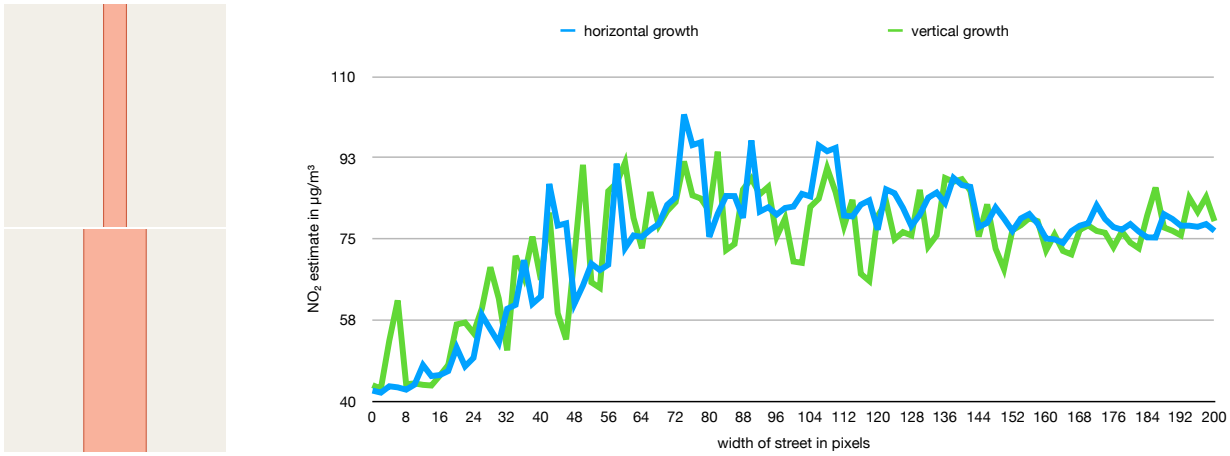
colored square images. On the other hand, the same images are then overlaid by the depiction of a motorway and a trunk road. We expect that different underlying entities provide different estimates according to the usual presence of sources for NO_2 pollution. We also expect an increase in the air pollution estimate whenever a road is added to the underlying entity. Depending on the type of road this increase might fluctuate. Table 3 shows the resulting pollution estimates by the CNN.

Different underlying entities do not lead to large differences in pollution estimates if there is no road. Only completely covering the image by a motorway or trunk road results in an estimate of over 45 $\mu g/m^3$. Additionally, trunk roads seem to have a much higher impact on the air pollution estimate than motorways. Adding a trunk road or motorway to any entity increases the air pollution estimate as expected. The amount of increase depends on the underlying entity of the map and what kind of road is present. This shows that the relationship of the entities that are visible in the map image are also important. There seem to be complex correlations between different entity features, which cannot be modeled easily in simpler models like linear regression.

Area Features: Area features describe *how large* a given entity is in the image. The area that an OpenStreetMap entity has on an image should contribute to the estimated pollution value. In this experiment, we use the trunk road entity to show the influence of the area. We build multiple images that contain a straight road that goes top to bottom or left to right through the center of the image. As the background we always use the same neutral background that depicts general land-use in OpenStreetMap. We then vary the width of that street either horizontally or vertically, depending on the street direction. A linear increase in the street's width is equivalent to a linear increase in the street's area.

Figure 6 shows some of the artificial OpenStreetMap images as well as a plot of MapLUR's output given the street width in pixels. As expected, an increasing width — and therefore an increasing area — of the street tends to increase the pollution estimate. Both horizontal and vertical growth have very similar curves that are not linear but instead seem to be more logarithmic. The similarity was expected, as during training, the images are augmented by rotation and flipping such that the direction of streets should not have any impact on the overall output.

Distance Features: Distance features describe *how far away* a given entity is from the image center. For this experiment, we create images that contain only one straight trunk road that is then moved vertically or horizontally, depending on the direction of the street. With this setup, we can control the distance of the motorway to the center of the image while fixing the area and entity features. We expect that the model produces higher estimates for images where the street is closer



(a) Two fake examples with a trunk road of width 10 px and 60 px, respectively.

(b) NO_2 estimate for a given width of the road.

Fig. 6. **Varying the width/area of the street while keeping other features such as distance to the center and type of street fixed.**

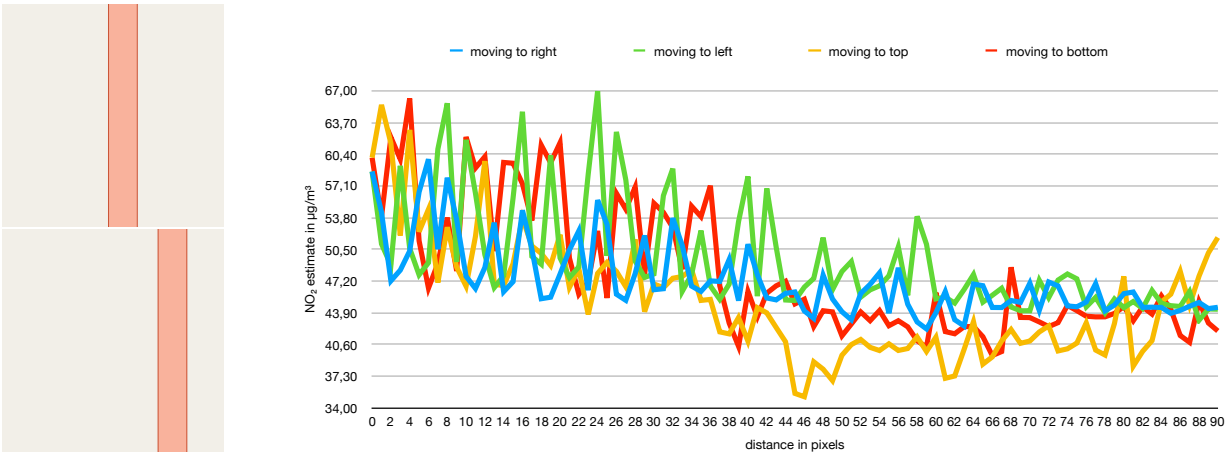
to the center, as this behavior was already observed in the guided backpropagation results. Also, the desired value from LAEI is coming from a 20 m subframe of the image which is in the image center. The model therefore should have learned a tendency to weight features from the center of the image more than from the borders.

Figure 7 shows image samples and the resulting estimation curves when moving the trunk road farther away from the center of the image. As expected, the proximity of a street to the image's center contributes to the overall NO_2 estimate positively. Pearson correlations of the distance with the estimated values are always lower than -0.6, indicating a relatively strong negative correlation. The curves that are shown are also not linear and can be better fitted by polynomials with a squared feature term than by a line. To capture this non-linearity, more sophisticated methods need to be used, which justifies the use of Random Forests or neural networks.

7 DISCUSSION

In this section, we discuss some advantages and some current limitations of our proposed paradigm and model. We believe that the advantages provide valuable additions to the models currently applied in land-use regression. Since this is the first paper applying a model based on our new paradigm, there are still some limitations regarding the applicability of our model to real world data sets, for which we provide some possible ways to overcome.

Interpretability. Firstly, purely data-driven models tend to be harder to interpret than simpler models, which is why they are often thought of as black-box models. This also raises the concern that the models may put too much focus on unreliable features that explain the specific data set well, but fail to generalize to other data sets. To alleviate these concerns, we have shown that it is possible to reveal the inner workings of the model, finding that the model's output heavily relies on land-use features such as streets or commercially used areas. These features are also commonly employed by traditional land-use regression models. We have also shown that MapLUR implicitly focuses on other commonly used land-use information such as distance and area features. This illustrates that purely data-driven approaches can yield interpretable models.



(a) Two fake examples with a trunk road that is moved by 10 px and 60 px, respectively.

(b) NO_2 estimate for a given distance from the center of the image to the trunk road.

Fig. 7. **Varying the distance of a trunk road to the image center pixel by pixel while keeping other features fixed.**

Feature extraction and complexity. In addition to information that closely resemble hand-crafted features, our model is able to extract signals from the image in an automated and optimized fashion. Thus, it can potentially capture more complex signals than modeled by hand-crafted features. For example, it is likely that features like curvature of streets, street signs, and traffic lights are also considered by MapLUR to estimate the air pollution. However, an analysis of these features remains future work. In particular, we believe that applying further model analysis techniques will enable researchers to find previously unknown features that can then be evaluated by experts and transferred to other, traditional land-use regression models. Thus, MapLUR's ability to extract interpretable features in combination with its inherent potential to model more complex relations of land-use and air pollution makes it a powerful tool for land-use regression.

Overfitting and generalization. Despite our strong results on the LAEI data set and gaining an intuition for how MapLUR works, we were not able to test generalizability across areas of interest due to the lack of similar data sets on different cities. While our setup allows for generalizability in principle, in practice, certain challenges may arise. In particular, deep learning methods are prone to overfitting, i.e, they may underperform when applied to input data that is very dissimilar to or not covered by the training data [17]. However, we suspect that this problem is less pronounced for MapLUR since map images are very structured and the model is therefore likely to see the vast majority of entity types that exist in the study area during training. Additionally, we want to stress again that previous models are often not applicable to new areas at all, since they often rely on data that is either not open or only locally available. Nonetheless, the model needs to be evaluated for every new application area before relying on its estimations. In order to use the model in new areas, it will usually be necessary to fit the model to some data from this area. Therefore, it is important to have training, validation, and test data sets that are similar in characteristics and representative of the whole study area. Typically, a simple random split is enough to achieve this [17].

Data Requirements and Application to Real World Data. MapLUR uses a CNN that contains a large number of weights due to its architecture. Training this deep-learning-based CNN requires more

data than regular land-use regression models, which is why we have evaluated the approach on data from a model [2] instead of real world data. We have shown that MapLUR works well given the 3,000 data points of LAEI's modeled NO_2 concentrations. Since this number is far greater than most real world data sets for land-use regression, future work needs to investigate the possibility of applying models based on the DOG paradigm in more realistic settings. We believe that one promising approach for research in this direction is the use of transfer learning, which has been shown to be an effective way of dealing with low-resource settings in both the areas of computer vision [38] and natural language processing [28]. Transfer learning could be applied to MapLUR or other models based on the DOG paradigm by pre-training the model on a large data set, like for example the LAEI data, and then fine-tuning it to a smaller data set of real world measurements. The global nature of features used in models based on the DOG paradigm ensures that this approach is generally possible. Additional large-scale data sets can also be collected in the context of mobile measuring campaigns [25, 41]. These data sets can then be used to provide further training data for the pre-training of DOG-based models.

Incorporating Distant Sources of Pollution. Our analysis has shown that using images depicting 80 m by 80 m areas for each data point leads to good results, as can be seen in Appendix B. However, previous approaches to air pollution estimation based on land-use regression have shown that it is useful to include information from wider surrounding areas in their features [49]. In the specific case of the MapLUR model, the size of the surrounding area that can be used is bounded by the resolution of the input images: If the area gets too large, the resolution of 224 x 224 px is not sufficient to encode the corresponding image. While this could be countered by increasing the input resolution, this would significantly increase the cost of training and prediction. Therefore, it is an interesting direction for further research to develop models that can take into account larger surrounding areas without increasing the image resolution. This could for example be achieved by using stacked convolutional neural networks or a combination of convolutional and recurrent neural networks.

Integration of Additional Data Sources. While we have focused only on map images as input, MapLUR was able to outperform all considered baselines. Nevertheless, previous work has shown that additional information can greatly improve the performance of air pollution models. One example of such data would be elevation maps [49], which can be integrated into MapLUR in a way similar to Experiment 4, where we provided the CNN with additional map image layers. Beyond this, there is a wide variety of methods to provide deep learning methods with additional information which holds great potential to further improve our results [4, 39].

8 CONCLUSION

In this paper, we have advocated DOG, a solely data-driven paradigm for air pollution estimation through land use regression. Models that follow this paradigm do not require manually engineered features and are based on data that is openly and globally available. This will ultimately result in models that are globally generalizable and can be applied in any area without modification. Working towards this goal, we have presented MapLUR, a deep learning based land-use regression model for air pollution estimation. We have shown that it can estimate NO_2 concentrations better than all considered baselines on a data set of modeled data from the Greater London area. While our analysis of MapLUR has shown that its data requirements are higher than commonly available data set sizes, we argued that transfer learning is a promising approach to alleviate this issue. We have also explored ways to analyze the factors that influence the prediction of this model, finding that a data-driven model architecture can be made interpretable by careful inspection of the trained model.

Thus, overall, this paper demonstrates the feasibility and advantages of our proposed data-driven paradigm DOG for land-use regression based air pollution modeling.

Future directions encompass work to further reduce the data requirements of data-driven models, the development of a comprehensive framework for extracting and interpreting features, as well as in-depth studies on real-world data, large-scale mobile measurements, and different cities.

ACKNOWLEDGMENTS

This work has been partially funded by the DFG grant "p2Map: Learning Environmental Maps - Integrating Participatory Sensing and Human Perception".

REFERENCES

[1] Matthew Adams. 2015. *Advancing the use of mobile monitoring data for air pollution modelling*. Ph.D. Dissertation. McMaster University, Hamilton.

[2] Air Quality Team (Greater London Authority). [n.d.]. London Atmospheric Emissions Inventory (LAEI). https://data.london.gov.uk/dataset/london-atmospheric-emissions-inventory-2013. Accessed: 2018-09-19.

[3] Md Saniul Alam and Aonghus McNabola. 2015. Exploring the modeling of spatiotemporal variations in ambient air pollution within the land use regression framework: Estimation of PM10 concentrations on a daily basis. *Journal of the Air & Waste Management Association* 65, 5 (2015), 628–640.

[4] Rahul Aralikatte, Heather Lent, Ana Valeria Gonzalez, Daniel Hershcovich, Chen Qiu, Anders Sandholm, Michael Ringaard, and Anders Søgaard. 2019. Rewarding Coreference Resolvers for Being Consistent with World Knowledge. http://arxiv.org/abs/1909.02392 cite arxiv:1909.02392Comment: To appear in EMNLP 2019.

[5] Yun Bai, Yong Li, Bo Zeng, Chuan Li, and Jin Zhang. 2019. Hourly PM2. 5 concentration forecast using stacked autoencoder model with emphasis on seasonality. *Journal of Cleaner Production* 224 (2019), 739–750.

[6] Yun Bai, Bo Zeng, Chuan Li, and Jin Zhang. 2019. An ensemble long short-term memory neural network for hourly PM2. 5 concentration forecasting. *Chemosphere* 222 (2019), 286–294.

[7] Rob Beelen, Gerard Hoek, Danielle Vienneau, Marloes Eeftens, Konstantina Dimakopoulou, Xanthi Pedeli, Ming-Yi Tsai, Nino Künzli, Tamara Schikowski, Alessandro Marcon, et al. 2013. Development of NO2 and NOx land use regression models for estimating air pollution exposure in 36 study areas in Europe–the ESCAPE project. *Atmospheric Environment* 72 (2013), 10–23.

[8] C Bonferroni. 1936. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze* 8 (1936), 3–62.

[9] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[10] Cole Brokamp, Roman Jandarov, MB Rao, Grace LeMasters, and Patrick Ryan. 2017. Exposure assessment models for elemental components of particulate matter in an urban environment: A comparison of regression and random forest approaches. *Atmospheric Environment* 151 (2017), 1–11.

[11] Bert Brunekreef and Stephen T Holgate. 2002. Air pollution and health. *The lancet* 360, 9341 (2002), 1233–1242.

[12] Alexander G Buevich, Alexander N Medvedev, Alexander P Sergeev, Dmitry A Tarasov, Andrey V Shichkin, Marina V Sergeeva, and TB Atanasova. 2016. Modeling of surface dust concentrations using neural networks and kriging. In *AIP Conference Proceedings*, Vol. 1789. AIP Publishing, 020004.

[13] David C Carslaw and Sean D Beevers. 2005. Estimations of road vehicle primary NO2 exhaust emission fractions using monitoring data in London. *Atmospheric Environment* 39, 1 (2005), 167–177.

[14] Alexandre Champendal, Mikhail Kanevski, and Pierre-Emmanuel Huguenot. 2014. Air pollution mapping using nonlinear land use regression models. In *International Conference on Computational Science and Its Applications*. Springer, 682–690.

[15] Ralph B d'Agostino. 1971. An omnibus test of normality for moderate and large size samples. *Biometrika* 58, 2 (1971), 341–348.

[16] Ralph B d'Agostino and Egon S Pearson. 1973. Tests for departure from normality. Empirical results for the distributions of b_2 and $\sqrt{b_1}$. *Biometrika* 60, 3 (1973), 613–622.

[17] Howard B. Demuth, Mark H. Beale, Orlando De Jess, and Martin T. Hagan. 2014. *Neural Network Design* (2nd ed.). Martin Hagan, USA.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

[19] Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. 1997. Support vector regression machines. In *Advances in neural information processing systems*. 155–161.

[20] Marloes Eeftens, Rob Beelen, Kees de Hoogh, Tom Bellander, Giulia Cesaroni, Marta Cirach, Christophe Declercq, Audrius Dedele, Evi Dons, Audrey de Nazelle, et al. 2012. Development of land use regression models for PM2. 5, PM2. 5 absorbance, PM10 and PMcoarse in 20 European study areas; results of the ESCAPE project. *Environmental science & technology* 46, 20 (2012), 11195–11205.

[21] Marloes Eeftens, Ming-Yi Tsai, Christophe Ampe, Bernhard Anwander, Rob Beelen, Tom Bellander, Giulia Cesaroni, Marta Cirach, Josef Cyrys, Kees de Hoogh, et al. 2012. Spatial variation of PM2. 5, PM10, PM2. 5 absorbance and PMcoarse concentrations between and within 20 European study areas and the relationship with NO2–Results of the ESCAPE project. *Atmospheric Environment* 62 (2012), 303–317.

[22] Derek M Elsom. 1992. *Atmospheric pollution: a global problem*. Blackwell Oxford.

[23] Junxiang Fan, Qi Li, Junxiong Hou, Xiao Feng, Hamed Karimian, and Shaofu Lin. 2017. A spatiotemporal prediction framework for air pollution based on deep RNN. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4 (2017), 15.

[24] Google LLC. 2018. Google Maps.

[25] David Hasenfratz, Olga Saukh, Christoph Walser, Christoph Hueglin, Martin Fierz, and Lothar Thiele. 2014. Pushing the spatio-temporal resolution limit of urban air pollution maps. In *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 69–77.

[26] Gerard Hoek, Rob Beelen, Kees de Hoogh, Danielle Vienneau, John Gulliver, Paul Fischer, and David Briggs. 2008. A review of land-use regression models to assess spatial variation of outdoor air pollution. *Atmospheric environment* 42, 33 (2008), 7561–7578.

[27] Gerard Hoek, Marloes Eeftens, Rob Beelen, Paul Fischer, Bert Brunekreef, K. Folkert Boersma, and Pepijn Veefkind. 2015. Satellite NO2 data improve national land use regression models for ambient NO2 in a small densely populated country. *Atmospheric Environment* 105 (2015), 173 – 180. https://doi.org/10.1016/j.atmosenv.2015.01.053

[28] Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. http://arxiv.org/abs/1801.06146 cite arxiv:1801.06146Comment: ACL 2018, fixed denominator in Equation 3, line 3.

[29] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[30] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[31] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-normalizing neural networks. In *Advances in neural information processing systems*. 971–980.

[32] Stéphane Lathuilière, Pablo Mesejo, Xavier Alameda-Pineda, and Radu Horaud. 2019. A comprehensive analysis of deep regression. *IEEE transactions on pattern analysis and machine intelligence* (2019).

[33] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[34] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[35] Xiaoli Li, Aorong Luo, Jiangeng Li, and Yang Li. 2019. Air Pollutant Concentration Forecast Based on Support Vector Regression and Quantum-Behaved Particle Swarm Optimization. *Environmental Modeling & Assessment* 24, 2 (01 Apr 2019), 205–222. https://doi.org/10.1007/s10666-018-9633-3

[36] Yuncheng Li, Jifei Huang, and Jiebo Luo. 2015. Using user generated online photos to estimate and monitor air pollution in major cities. In *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*. ACM, 79.

[37] Wu Liu, Xiaodong Li, Zuo Chen, Guangming Zeng, Tomás León, Jie Liang, Guohe Huang, Zhihua Gao, Sheng Jiao, Xiaoxiao He, et al. 2015. Land use regression models coupled with meteorology to model spatial and temporal variability of NO2 and PM10 in Changsha, China. *Atmospheric Environment* 116 (2015), 272–280.

[38] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. 2018. Exploring the Limits of Weakly Supervised Pretraining. http://arxiv.org/abs/1805.00932 cite arxiv:1805.00932Comment: Technical report.

[39] Julian McAuley and Jure Leskovec. 2012. Image labeling on a network: using social-network metadata for image classification. In *European conference on computer vision*. Springer, 828–841.

[40] Xia Meng, Li Chen, Jing Cai, Bin Zou, Chang-Fu Wu, Qingyan Fu, Yan Zhang, Yang Liu, and Haidong Kan. 2015. A land use regression model for estimating the NO2 concentration in Shanghai, China. *Environmental research* 137 (2015), 308–315.

[41] Denise R Montagne, Gerard Hoek, Jochem O Klompmaker, Meng Wang, Kees Meliefste, and Bert Brunekreef. 2015. Land use regression models for ultrafine particles and black carbon based on short-term monitoring predict past spatial variation. *Environmental science & technology* 49, 14 (2015), 8712–8720.

[42] David W Morley and John Gulliver. 2018. A land use regression variable generation, modelling and prediction tool for air pollution exposure assessment. *Environmental Modelling & Software* 105 (2018), 17–23.

[43] Sheena Muttoo, Lisa Ramsay, Bert Brunekreef, Rob Beelen, Kees Meliefste, and Rajen N Naidoo. 2018. Land use regression modelling estimating nitrogen oxides exposure in industrial south Durban, South Africa. *Science of the Total Environment* 610 (2018), 1439–1447.

[44] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 807–814.

[45] OpenStreetMap contributors. 2018. Apache module mod_tile. https://github.com/openstreetmap/mod_tile.

[46] OpenStreetMap contributors. 2018. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org.

[47] EG Ortiz-García, S Salcedo-Sanz, ÁM Pérez-Bellido, JA Portilla-Figueras, and L Prieto. 2010. Prediction of hourly O3 concentrations using support vector regression algorithms. *Atmospheric Environment* 44, 35 (2010), 4481–4488.

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

[49] Patrick H Ryan and Grace K LeMasters. 2007. A review of land-use regression models for characterizing intraurban air pollution exposure. *Inhalation toxicology* 19, sup1 (2007), 127–133.

[50] Vikas Singh. 2016. Higher Pollution Episode Detection Using Image Classification Techniques. *Environmental Modeling & Assessment* 21, 5 (01 Oct 2016), 591–601. https://doi.org/10.1007/s10666-015-9497-8

[51] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806* (2014).

[52] Meng Wang, Rob Beelen, Tom Bellander, Matthias Birk, Giulia Cesaroni, Marta Cirach, Josef Cyrys, Kees de Hoogh, Christophe Declercq, Konstantina Dimakopoulou, et al. 2014. Performance of multi-city land use regression models for nitrogen dioxide and fine particles. *Environmental health perspectives* 122, 8 (2014), 843.

[53] Kathrin Wolf, Josef Cyrys, Tatiana Harciníková, Jianwei Gu, Thomas Kusch, Regina Hampel, Alexandra Schneider, and Annette Peters. 2017. Land use regression modeling of ultrafine particles, ozone, nitrogen oxides and markers of particulate matter pollution in Augsburg, Germany. *Science of the Total Environment* 579 (2017), 1531–1540.

[54] Jiansheng Wu, Jiacheng Li, Jian Peng, Weifeng Li, Guang Xu, and Chengcheng Dong. 2015. Applying land use regression model to estimate spatial variation of PM2. 5 in Beijing, China. *Environmental Science and Pollution Research* 22, 9 (2015), 7045–7061.

[55] Jingjing Xie, Xiaoxue Wang, Yu Liu, and Yun Bai. 2018. Autoencoder-based deep belief regression network for air particulate matter concentration forecasting. *Journal of Intelligent & Fuzzy Systems* 34, 6 (2018), 3475–3486.

[56] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853* (2015).

[57] Chao Zhang, Junchi Yan, Changsheng Li, Hao Wu, and Rongfang Bie. 2018. End-to-end learning for image-based air quality level estimation. *Machine Vision and Applications* 29, 4 (2018), 601–615.

Fig. 8. **LAEI cells sampled for the experiments.** The 3,000 blue cells are the training data set and the 1,500 red cells are the test data set. (Underlying OpenStreetMap Image: © OpenStreetMap contributors)

A SAMPLED LAEI CELLS

Figure 8 depicts the cells which we sampled from LAEI. The blue cells are used for training our models while the red cells are used to evaluate model performance for unseen locations.

B ANALYSIS OF AREA SIZE

In this work, we only used 80 m by 80 m images as inputs for our model. However, despite the potential evaluation issues with overlapping images described in Section 3.1, it is still interesting to see how our model behaves when it is able to see more or less of the surroundings. Therefore, the model was provided with OpenStreetMap images depicting square areas around the data point with side lengths of 60 m, 80 m, 100 m, 200 m, 500 m, and 1000 m while maintaining a resolution of 224 px by 224 px. The mean results after 40 evaluations can be seen in Table 4.

Our model does not benefit from the increased image size as can be seen from both R^2 and RMSE. The mean performance decreases consistently with each increase in depicted area size over 80 m by 80 m. This implies that the potential evaluation issue with overlapping images, which is described in Section 3.1, is not very severe since the model should be gaining performance with larger images otherwise. It also suggests that the very close surroundings are important and that information from further away is not helping. However, the model is suffering from performance loss with smaller areas than 80 m by 80 m. Thus, it seems that a side length of 80 m for the depicted square areas is optimal for MapLUR especially considering the fact that all other results are significantly different according to the Wilcoxon signed-rank test even after Bonferroni correction, since $p < \frac{0.05}{n}$ for $n = 9$.

Table 4. **Evaluating the influence of the area size depicted in each map image on MapLUR's performance.** The results suggest that 80 m by 80 m areas are optimal.

Model	R^2	RMSE
Mean baseline	0.000	13.971
Linear regression	0.487	10.004
Multi-layer Perceptron	0.499	9.887
Random Forest	0.662	8.119
60 m	0.626	8.511
80 m	**0.673**	**8.002**
100 m	0.637	8.381
200 m	0.618	8.603
500 m	0.597	8.833
1000 m	0.390	10.856

C EVALUATION METRICS

Given the desired target values $y = \{y_1, y_2, \ldots, y_n\}$ and the model's output $\hat{y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$, two commonly used metrics in land-use regression papers, namely R^2 and root-mean-square error (RMSE) [14, 21, 37], are used to evaluate the model on the evaluation set.

On the one hand, R^2 describes how much of the target's variation is explained by the model:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2},$$

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ is the mean of all desired target values. The metric can take values from $-\infty$ to 1. A R^2 of 1 indicates a perfect fit. A value of 0 is achieved by always estimating the mean of the evaluation set's target values. Negative values indicate that the model is worse than always estimating the mean.

On the other hand, the RMSE is, as the name already suggests, the square root of the mean of the squared errors:

$$\mathrm{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}.$$

Thus, RMSE can only take non-negative values, where 0 would be perfect for this metric and larger RMSEs are worse.

D MODEL BUILDING PROCEDURES FOR THE BASELINES

The following explains the model building procedures for the baseline methods in more detail.

D.1 Linear Regression

The first baseline model we consider is the commonly used linear regression. The model's development is based on a supervised stepwise selection which was used in the Escape project [20] for land-use regression model development before. Each predictor variable is ranked based on the

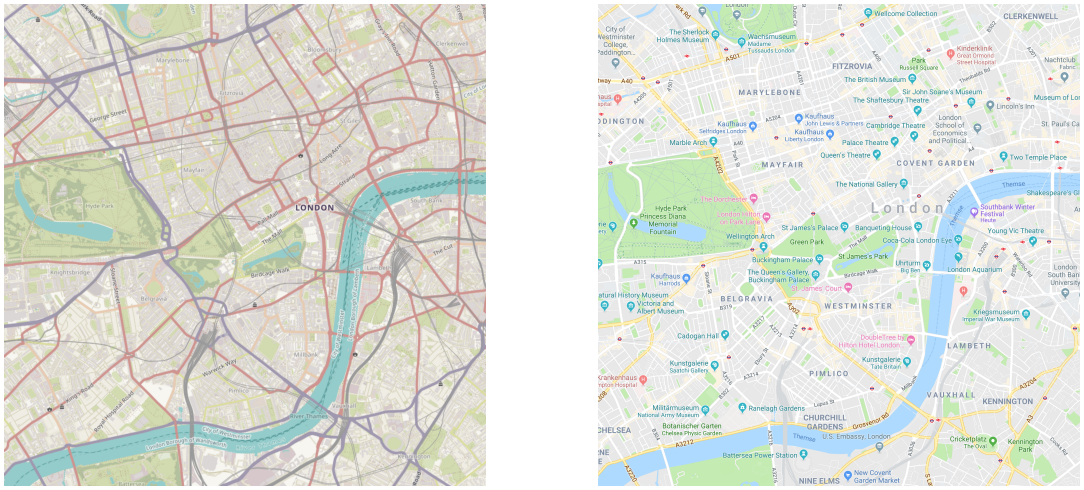
Table 5. **Random Forest hyperparameters optimized using stochastic search with the corresponding search spaces.**

Hyperparameter	Search space
Number of trees	1 to 1000
Fraction of features to consider at most per split	0.0 to 1.0
Minimum samples required to be a leaf node	1 to 100
Minimum samples required to split a node	2 to 20
Build trees with bootstrap samples	True or False

model's adjusted R^2 from a univariate regression. The adjusted R^2 used by Eeftens et al. [20] is like the R^2 but penalizes adding variables which do not fit the model. Thus, ideally only independent variables which affect the dependent variable are used. If there are variables that are of the same category but with different buffer sizes, then only the variable with the highest score is considered for use in the final model due to the high correlation of these variables between each other. The model starts with the variable that achieved the highest score. Thereafter, each one of the remaining variables is temporarily added to the model, evaluated, and the best performing variable is added to the model permanently if it increases the model's adjusted R^2 by at least 0.01. This is repeated until no variables are left. Then all selected variables with a p-value greater than 0.1 are removed and the resulting model is fit again, just like described in Eeftens et al. [20]. Finally, the variance inflation factors (VIFs) are calculated for each variable in order to quantify the increase in variance due to collinearity of the variables. If a variable has an VIF that is greater than 3, the variable with the largest VIF is removed and the model is refit. In accordance with Eeftens et al. [20] this is also repeated until no variable has an VIF greater than 3.

D.2 Random Forest

Another baseline is a Random Forest model which employs ensembles of decision trees for its estimations [9]. This model is built in a similar way to the procedure in Brokamp et al. [10]. The steps of the model building procedure are described in the following. First, the best buffer radii for each type of variable were determined based on the adjusted R^2 of a univariate regression on the training data set. As described in Section D.1 before, the adjusted R^2 is like the R^2 but penalizes adding variables which do not fit the model. Then, an initial Random Forest is considered for the following which uses values that have shown to work decently in preliminary experiments: It builds 500 trees, considers half the available features when looking for the best split, and uses the default values of scikit-learn's Random Forest implementation for the other hyperparameters [48]. The best variables of each type are then fitted using this Random Forest to rank the variables based on the variable importance score of the Random Forest. Thereafter, the least important variables are removed iteratively. For each iteration, the model is fitted to the remaining variables on the training data set and the out of bag R^2 is calculated by estimating each training sample without using the trees that had the training sample in their bootstrap sample. This metric can be used with Random Forests to estimate performance without an independent test set. The set of variables that achieved the best performance is selected for further use. Then the best hyperparameters for the Random Forest are found by a stochastic search that samples hyperparameter values and evaluates them with a ten fold cross-validation on the folds of the training set. Each hyperparameter value is sampled from a uniform distribution with a specific search space. Table 5 shows these



(a) OpenStreetMap images (© OpenStreetMap contributors)

(b) Google Maps images (Map data: Google)

Fig. 9. **Comparison of OpenStreetMap and Google Maps images.** OpenStreetMap images contain more color-coded entities such as streets. As the visual style is highly important for the CNN to learn the task, the simpler Google Maps style produces worse results.

hyperparameters with their search spaces. This search ran for three hours and used six CPU cores of an Intel Xeon E5-2690V4 processor to fit as many models as possible. During this search, 1,051 sets of hyperparameters were evaluated. The best hyperparameters are then used to fit a Random Forest on the complete training set and evaluate the model on the test set.

D.3 Multi-layer Perceptron

For the last baseline, we employ a multi-layer Perceptron, which is a type of neural network. For this model, we base our model building procedure on the one used by Alam and McNabola [3]. We adapt it slightly by first selecting the best buffer radius for each variable type based on the adjusted R^2 of an univariate regression. We use the adjusted R^2 again for this selection to be consistent with the model building procedures for linear regression and Random Forest described before. Then we search for the best performing architecture by evaluating models with different number of hidden layers and neurons for each layer with ten fold cross-validations on the training data set. This is done by randomly sampling the number of layers from a uniform distribution ranging from 1 to 3 hidden layers and randomly sampling the number of neurons for each layer from another uniform distribution which ranges from 1 to 30. These bounds are chosen since they encompass the architectures of all previously published models that we found [1, 3, 12, 14, 37]. We randomly sample model architectures and evaluate them for 24 hours during which 2,629 architectures were evaluated. The best performing architecture is then trained on the complete training set and evaluated on the test set.

E COMPARISON OF OPENSTREETMAP AND GOOGLE MAPS IMAGES

Figure 9 compares Google Maps and OpenStreetMap images, showing the differences between both default styles shown on the services' websites. While OpenStreetMap provides at least four colors to denote different types of streets, Google Maps only uses two. Google Maps does not color-code all of the available information to ease the visual effort of the user. This however is not helpful for the CNN model, which works better with clear visual cues that denote entities.

Evaluating the multi-task learning approach for land use regression modelling of air pollution

Andrzej Dulny[1,*], Michael Steininger[1], Florian Lautenschlager[1], Anna Krause[1] and Andreas Hotho[1]

[1]University of Würzburg, Germany

*E-mail: andrzej.dulny@stud-mail.uni-wuerzburg.de

Abstract. Air pollution has been linked to several health problems including heart disease, stroke and lung cancer. Modelling and analyzing this dependency requires reliable and accurate air pollutant measurements collected by stationary air monitoring stations. However, usually only a low number of such stations are present within a single city. To retrieve pollution concentrations for unmeasured locations, researchers rely on land use regression (LUR) models. Those models are typically developed for one pollutant only. However, as results in different areas have shown, modelling several related output variables through multi-task learning can improve the prediction results of the models significantly. In this work, we compared prediction results from singletask and multi-task learning multilayer perceptron models on measurements taken from the OpenSense dataset and the London Atmospheric Emissions Inventory dataset. LUR features were generated from OpenStreetMap using OpenLUR and used to train hard parameter sharing multilayer perceptron models. The results show multi-task learning with sufficient data significantly improves the performance of a LUR model.

1. Introduction

Evidence suggests that air pollution has adverse effects on health [1, 2, 3] and the environment [4]. In order to research, assess and prevent these effects, availability of high quality measurements of air pollutants is necessary. Official authorities maintain stationary air quality monitoring networks, mostly equipped with sensors for multiple pollutants [5, 6]. However, such stations only offer data from a limited number of locations, as usually only few stations are present within a single city and land-use regression (LUR) models have been used successfully to account for spatial variability within cities and for epidemiological analyses [7]. While LUR models are trained on only one pollutant, several research areas have shown the potential of training on multiple related target variables, so called multi-task learning [8].

Air pollution monitoring stations often measure concentrations of more than one pollutant and the high spatial and temporal correlation of air pollutant concentrations [9] suggests that the emissions of different pollutants depend on the same set of factors. Thus, the tasks of modelling the pollutants should be highly related and modelling them with a multi-task learning model might improve the accuracy of the predictions. However, this approach has not been assessed yet in the context of modelling air pollution. To evaluate it, the performance of a multilayer perceptron LUR model is compared between single-task and multi-task learning on two different datasets - measurements taken from the OpenSense project collected by low-cost, portable sensors in the city of Zurich and modelled concentrations taken from the London Atmospheric Emissions Inventory.

The contribution of this work is twofold: We (i) propose a new approach to developing LUR models using concentration data from multiple pollutants, which takes advantage of the available measurements as shown in figure 1 and (ii) demonstrate the potential of the multi-task learning approach compared to traditional single-task models for LUR.

The work is structured as follows: In Section 2 we summarize the related work. Section 3 describes the air pollution datasets and LUR features used to develop the models. The selected multi-task learning framework is described in Section 4 and the experimental approach and the models in Section 5. In Section 6 we present our results and in Section 7 we discuss the advantages and limitations of the multi-task learning approach. Section 8 provides a conclusion and outlook for future work.

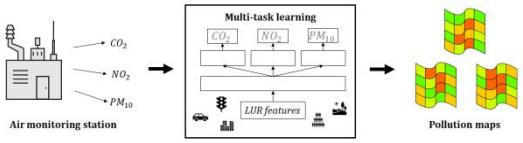


Figure 1. Air monitoring stations collect data from different pollutants. In our approach we use multi-task learning to improve the accuracy of pollution maps.

2. Related work

2.1. Land-Use regression

LUR models are an active field of research as the public awareness of the health and environmental effect of air pollution grows. Such models have been developed for numerous large cities worldwide. A 2008 review collected models developed for several cities in Europe as well as the USA [10] and a recent review from 2017 includes LUR models for 16 different cities worldwide [11]. Within the European ESCAPEProject aimed at assessing the long-term effects of air pollution on human health, models have been developed for 36 cities in Europe using a standardized approach of model selection for a linear regression [1].

Traditionally, linear regression has been used for LUR [10, 11], however, several other machine learning models have been proposed to increase the accuracy of the predictions and model non-linear relationships between the variables. Generalized additive models are one such example and they have been used to improve prediction scores in LUR models of nitrogen oxides (NO_x) in Southern California [12] and $PM_{2.5}$ models in Beijing-Tianjin-Hebei (BTH) region in China [13]. Brokamp et al. used random forest regression to improve concentration predictions of in the urban city of Cincinnati, Ohio [14], and in [15] this approach is used to model NO_2 concentration in a metropolitan area of Japan.

Models using neural networks for LUR have also been proposed. For example, Alam and McNabola compare linear LUR models with multilayer perceptron models, achieving better results with the latter [16], while Adams and Kanaroglou use multilayer perceptron LUR models to construct real-time air pollution health risk maps [17].

Steininger et al. use a deep learning neural network to model air pollutant concentrations directly from globally available map images [18] and Lautenschlager et al. use features generated from geographical information available in the OpenStreetMap database [19] to develop models performing better than similar models using features from local or closed sources [20].

In all of these studies, only one pollutant is predicted with a single model. It has been shown that different air pollutants show high temporal and spatial correlation patterns [21] and thus, the tasks of modelling different air pollutants can be highly related. The goal of this work is to explore the possibility of achieving better prediction results using a multi-task learning framework.

2.2. Multi-task learning

Multi-task learning is a machine learning paradigm in which several related tasks are modelled simultaneously. A shared representation is used to guide the models to the most relevant features, thus potentially improving generalization and performance [8, 22]. It has been shown to increase

effectiveness of machine learning models in a wide range of fields.

Collobert and Weston proposed a multi-task learning approach for natural language processing, in which several speech related predictions are made using a single neural network [23]. Gibert et al. use a multi-task learning framework to automatically detect anomalies for railway track inspections using machine vision. The multi-task model performs with increased accuracy as compared to single-task detectors [24].

Ramsundar et al. use a multi-task framework to develop large-scale models in the field of drug discovery. The results show increasing prediction accuracy when additional tasks are added to the model and the shared representation learned by the models can be transferred to other tasks, which were not used during training [25].

Caruana [8] explored the direct comparison of single-task models and multi-task models, the latter achieving better results on problems including autonomous driving simulations, recognizing knobs on images of doors and predicting the severity of pneumonia.

Multi-task learning has been applied in several fields where multiple related tasks are modelled, performing better than using single-task models separately. However, its application in the context of air pollution modelling has not been assessed. This work is aimed at filling this gap, by comparing single-task LUR models with models used to predict several air pollutants at the same time.

We used a multilayer perceptron hard parameter sharing model for multi-task learning, as it is the most commonly used approach in other applications and because it allows for a direct comparison within a single framework.

3. Materials

In this section, the data sources used for the evaluation of the multi-task learning approach are introduced: the OpenSense dataset collected during a mobile sensing campaign in Zurich [26] and the London Atmospheric Emission Inventory [27], which contains a dataset developed using an atmospheric dispersion model and is published by London authorities. Furthermore, the LUR features which have been used to develop single-task and multi-task learning models are discussed in this section.

3.1. OpenSense dataset

The OpenSense Project collected pollution data over the period of several years between 2012 and 2016 from mobile, low-cost sensing units equipped with an ultrafine particle (UFP) sensor, carbonmonoxide (CO) sensor and ozone (O_3) sensor placed on top of ten street cars, travelling on regular routes within the city of Zurich. The particulate matter pollution was sampled every 5 s and the O_3 and CO concentrations every 20 s [28]. A GPS signal receiver provided spatial information about the measurements. The gas sensors were equipped with water and dust covers to minimize possible interference [28].

3.1.1. Data selection

For creating the LUR models, measurements from the year 2014 have been selected from the dataset. Although there are certainly LUR models being developed for smaller time scales using additional weather information as features ([29], [30]), the most common approach is to consider a long time period for averaging the measurements. This removes any possible seasonal trends, which have a considerable influence on air pollution [31, 32]. Additionally, aggregated means are important from a regulatory perspective, as for example the European Commision enforces limits on annual averages for emissions of air pollutants [33]. We used the concentration data collected during the year 2014, with the exception of CO, where measurements were not available for the first two months of 2014. Instead, to maintain a comparable representation of all seasons, CO was averaged using the period of one year starting from 03/2014. Table 1 summarizes the data that has been selected from the OpenSense dataset to develop the models.

Table 1. Subset of the OpenSense dataset considered.

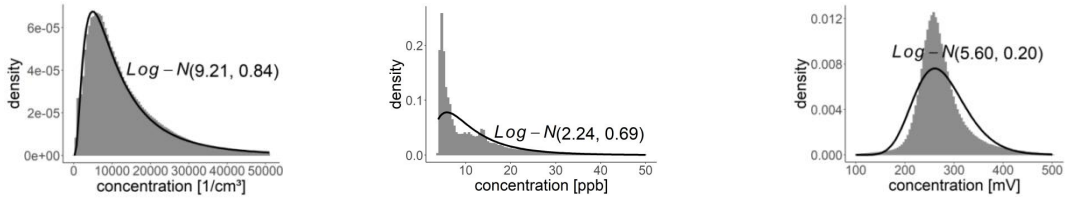
Pollutant	Start	End	Samples
UFP	01/2014	12/2014	11.9 Mio
O_3	01/2014	12/2014	5.3 Mio
CO	03/2014	02/2015	19.7 Mio

3.1.2. Data preprocessing

The UFP dataset has been properly calibrated and filtered and thus contains accurate measurements [34]. Reference measurements and internal variables of the sensors are not available for the ozone and carbonmonoxide datasets and thus a null-offset calibration cannot be done. The concentration measurements for those two pollutants are therefore taken from the factory pre-calibrated sensors without additional calibration. Following Hasenfratz et al., an initial GPS-filter is applied to assure an accurate geo-tagging of the concentrations, based on the horizontal dilution of precision (HDOP) [29]. Hasenfratz et al. discarded all measurements with a HDOP of smaller than 3. To obtain an even more accurate positioning we used the threshold of 2. Additionally, all measurements taken outside of the boundaries of the routes taken by the street cars are also discarded. The remaining samples (97.1% of the UFP measurements, 96.8% of the CO measurements and 96.9% of the O_3 measurements) were used for further processing.

3.1.3. Data aggregation

Following [29], where a 100 m×100 m grid was used to develop a LUR model with the OpenSense UFP data, annual averages for the same spatial resolution of 100 m were calculated. Because the measurements were taken by mobile sensors, there was a considerable variation in the number of observations in each cell, ranging from 1 to over 300 000 for the 100 m grid. To ensure that the mean annual concentrations are reliable and to exclude possible outliers due to positioning errors, cells with less than 50 measurements were discarded.



(a) Ultrafine particles (b) Ozone (c) Carbonmonoxide

Figure 2. Distribution of the concentrations of ultrafine particles, ozone and carbon monoxide in the year 2014 as compared to a log-normal distribution. Based on the OpenSense dataset.

3.1.4. Data evaluation

Data quality for air pollutants can be assessed using a well established observation that the measurements of air pollutants approximately follow a log-normal distribution [35]. Figure 2 shows the empirical distribution of the raw measurements in comparison to a theoretical log-normal distribution with the same mean and standard deviation. The relatively close fit of both distributions for the UFP indicates that the measurements are reliable. This does not hold true for the CO and O_3 data, where there are substantial deviations from the log-normal distribution. The poor fit of the log-normal distribution function to the concentration measurements for CO and O_3 indicates poor data quality, which can be attributed to the sensors not being adequately calibrated. A proper calibration would require accurate reference data for a wide range of humidity and weather conditions, which is not available for this dataset. The variability of atmospheric conditions in which the mobile sensors have been used can thus result in the measurements not being accurate [34].

3.1.5. Summary

The previous analysis of the OpenSense dataset suggests that the measurements of CO and O_3 might be noisy and the question of whether an evaluation of multi-task learning can be done on such a

dataset should be addressed. While a dataset containing measurements from well calibrated sensing units would serve this purpose better, to our best knowledge there is no such a dataset that also contains enough samples to enable training multilayer perceptron models. The measurements of UFP from the OpenSense dataset have been used successfully to develop LUR models before [29, 19] and for this reason, while accepting the limitations of using noisy measurements of the other pollutants, we decided to include the OpenSense dataset in our analysis.

The OpenSense datasets that have been used here can be accessed online: [36] for the UFP dataset and [37] for the CO and O_3 datasets.

3.2. London Atmospheric Emissions Inventory

The London Atmospheric Emissions Inventory (LAEI) is a data collection containing estimates of pollutant emissions and their sources for a given year in the city of London. The input factors include traffic data from road and rail networks, domestic and commercial fuel consumption, aviation, and pollution from individual industrial sites. The emission data is used to model ground-level average yearly concentrations of air pollutants on a 20 m × 20 m grid using a atmospheric dispersion model.

In this work, we used the 2013 version of the inventory to develop LUR models for multiple pollutants: nitrogen dioxide (NO_2), nitrogen oxides (NO_x), particulate matter of diameter less than 10 μm (PM_{10}), number of days with a daily mean PM10 concentration greater than 50 μg m^{-3} (PM_{10d}), and particulate matter of diameter less than 2.5 μm ($PM_{2.5}$). It is important to stress that the LAEI contains modelled annual mean concentrations and not measurements from air monitoring stations. However, the inventory has been used for LUR modelling, as for example Steininger et al. developed deep learning LUR models using the concentration data for NO_2 from the LAEI [18].

3.3. Features

We generated LUR features using the OpenLUR approach [19]. Starting from a given point, the total area of commercial, industrial and residential buildings within different radiuses (50m – 3000 m in 50m steps) was computed using geographical information from OpenStreetMap. Additionally, the total length of roads of different types and the distance to the closest traffic signal, motorway, primary road and industrial area was calculated. In total 244 features were generated this way.

Table 2. Results of feature selection for both datasets.

(a) Features selected for the OpenSense data	(b) Features selected for the LAEI data
Features selected	Features selected
residential area within 1550m	length of large roads within 50m
distance to the closest primary road	residential area within 2150m
length of large roads within (100m, 850m, 1500m)	distance to the closest traffic signal
distance to the closest industrial area	
residential area within (700m, 2950m, 3000m)	
industrial area within (1750m, 2550m, 3000m)	
commercial area within 3000m	
industrial area within	

3.3.1. Selection

Feature selection is a systematic method of selecting the variables upon which to build the model. Selecting only the relevant features ensures that the model is easily interpretable and improves the performance of the model by enhancing generalization [38]. We used a selection method based on the best performing features on linear models, similar to [1]. Starting from an empty model the feature improving the average R^2 score the most is iteratively added to the model until an improvement threshold of 1% is no longer reached.

Features were selected separately for the OpenSense and LAEI. In total 13 features have been selected for the OpenSense data at and 3 features for the London Atmospheric Inventory as shown in table 2.

4. Method

In this work, a hard-parameter sharing multi-task learning approach is implemented using a multilayer perceptron model with two hidden layers. Multilayer perceptron models have been applied successfully in LUR to model single pollutants [16]. It allows for a straightforward translation into a multi-task learning framework by providing additional outputs and thus a relatively direct comparison. When hidden layers are shared between outputs, the network is forced to learn a shared representation between the tasks which reduces the risk of overfitting [39, 22], possibly improving the performance of the model.

4.1. Network structure

For a direct comparison between multi-task learning and single-task learning, the model's overall structure is kept constant while varying the number of shared layers. This creates three distinct models with similar architecture but different degrees of multi-task learning: a fully multi-task learning model when all of the hidden layers are shared between pollutants, a single-task model if all of the layers are task-specific, as well as an intermediate model with one shared layer and one task-specific layer.

4.2. Training

All training of the multilayer perceptron models was performed using the Keras library version 2.3.1 [40]. The Adam optimizer was used for the weight updating with the default learning rate of 0.001. The mean squared error is used as the loss function and additionally the R^2 score is monitored. An early stopping condition is used to determine the end of the training before the maximum number of epochs set to 2000. After each epoch, the improvement of the mean squared error on the validation data is checked and if the score did not improve during the last 20 epochs the training stops and the best performing weights are restored. The final R^2 score of the model on the validation data is calculated and averaged over all cross-validation sets. The same cross-validation division was used for training and scoring all models. This score is then used to select the best performing model.

5. Experiments

In order to evaluate the multi-task learning approach the data is split into several training and evaluation sets and used to train the baseline and experimental models. This procedure and the baseline models are described in the following.

5.1. Data selection

In total, 929 cells with annual mean concentrations were available from the OpenSense dataset and 5851915 from the LAEI. However, only 4500 cells were sampled from the large dataset for training and evaluating the models and the features have been calculated only for those measurements. The decision to only include a limited number of data points was made due to the high computational cost of obtaining LUR features from OpenStreetMap and to increase the generalizability of our evaluations, as datasets usually used for LUR only contain limited number of samples [29]. For a comprehensive evaluation of the multi-task learning approach, training sets of different sizes were included. For the OpenSense dataset samples of 100, 300 and 500 measurements were sampled uniformly as training sets to investigate the influence of the size of the training data on the performance of the multi-task learning models. The models for the London Atmospheric Emissions were trained using sample sizes of 100, 300, 500 and 3000. The measurements not included in the training set were used to evaluate the resulting models to obtain the final score. All model types (including the baseline models) were trained using the same training set and evaluated using the same test set and used the same cross-validation division for all models. In total 7 different training sets were created this way.

5.2. Baseline

In order to evaluate the multi-task learning model and put the observed differences in context, the LUR models for the available datasets are first developed using traditional approaches - linear

regression and random forest regression. The details on how models were trained and evaluated using both approaches are provided here.

5.2.1. Linear model

Linear regression has been traditionally used in LUR models [41], it is therefore a good baseline to consider for the performance of other models. For each dataset, a linear model was fitted on the training set using the features selected with the algorithm described in Section 3.3.1. Each of the pollutants was modelled separately. The resulting models were then evaluated on the available test samples.

5.2.2. Random forest

Random forest regression has been used for LUR models yielding good prediction results [14], which is the reason why it was included as a baseline. For each dataset, a random hyperparameter search was performed to find the optimal number of trees (10 – 2000), fraction of features considered (0.0 – 1.0), fraction of data considered (0.0 – 1.0) and minimum samples for a split (2 - 21). The remaining hyperparameters for the model use the default values provided by the scikit-learn library in version 0.22.1. The mean R^2 score from a ten-fold cross-validation was used to select the best performing model, which was then fitted on the whole training set and evaluated using the test set.

5.3. Hyperparameters for multi-task models

To evaluate the multi-task learning models, a hyperparameter optimization procedure was implemented for each of the training sets to find the best performing models of each structure. All models have two hidden layers, each of which contains the same number of neurons. They differ only in the number of layers shared between the different pollutants.

Hyperparameter optimization was performed using random search for the number of neurons per layer (5 - 200), dropout rate (0 - 0.8) and L_2-regularization (0.0001 - 1). For other hyperparameters, the default values provided by the Keras library version 2.3.1 were used during the training [40]. All three models with different degrees of multi-task learning were trained using this set of hyperparameters and evaluated using a ten-fold cross-validation method, similarily to the training of the random forest regression models. The subsample of the training set left-out by the given crossvalidation iteration is used as validation data for monitoring the performance of the model during training and for calculating the final score.

6. Results

In this chapter we present the results of the different LUR models. This includes the results of the baseline models and the comparison of different multi-task learning models and single-task learning models. The models were trained using the best found hyperparameters and evaluated using the test dataset which was not used before. The same tenfold cross-validation division of the training set used during hyperparameter search was applied to keep track of the R^2 score during training for the purpose of early stopping. Each cross-validation was performed 30 times in total. This was a compromise between the high computational cost of fitting the models and the requirements for an accurate estimate of the scores.

6.1. OpenSense dataset

6.1.1. Multi-task learning

Table 3 shows the average R^2 scores of models using different degrees of multi-task learning and single-task learning on the OpenSense dataset. Zero shared layers correspond to single-task learning, while with two or one shared layers features and activations of hidden layers are shared between pollutants thus corresponding to multi-task learning.

The bold scores in table 3 indicate the best model for each training sample. The results show an improvement of the R^2 scores by using at least some shared representation as compared to single-task

learning for all training samples considered.

Table 3. Average R^2 scores on the test samples from the OpenSense dataset using multilayer perceptron models with different numbers of shared layers. The increase is calculated between the single-task learning model (zero shared layers) and the best performing multi-task learning model (at least one shared layer).

	Shared layers			
Samples	0	1	2	Increase
100	0.224	0.169	**0.224**	+0.41%
300	0.410	**0.448**	0.391	+9.23%
500	0.463	**0.474**	0.379	+2.26%

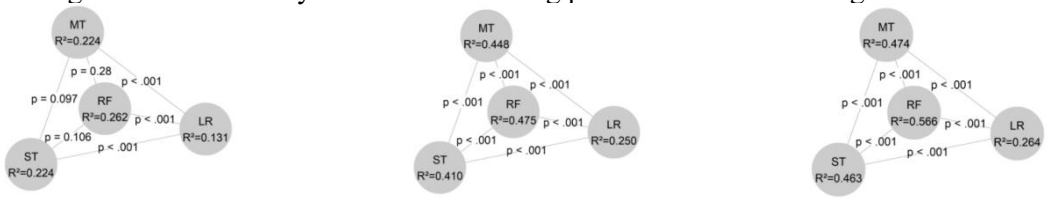
The optimal structure of the model varies with the training sample considered as does the amount of improvement as shown in the increase percentage of the R^2 scores in table 3. The one-way ANOVA performed for each training set shows that the modelling approaches differ significantly ($p < .001$).

Table 4. Average R^2 scores on the test samples from the OpenSense dataset using linear regression (LR), random forest regression (RF), as well as single-task learning (ST) and multi-task learning (MT) using a multilayer perceptron (MLP).

			MLP	
Samples	LR	RF	ST	MT
100	0.131	0.262	0.224	0.224
300	0.250	0.475	0.410	0.448
500	0.264	0.566	0.463	0.474

6.1.2. Comparison

The comparison of the results of both baseline models, single-task learning models and the best multi-task learning models for the OpenSense dataset is shown in table 4. The best performing model with at least one shared layer has been taken to represent the multi-task learning approach. To check whether the resulting R^2 scores were significantly different, for each training sample the models were tested pairwise using the Mann-Whitney U-test. The resulting p-values are shown in figure 3.



(a) 100 training samples (b) 300 training samples (c) 500 training samples

Figure 3. Pairwise Mann-Whitney U-tests between linear regression models (LR), random forest models (RF) and multilayer perceptron models using single-task (ST) and multi-task learning (MT) on the Opensense dataset.

For 300 and 500 training samples, the random forest model performs significantly better than any other model. The linear models perform significantly worse than non-linear models. For all considered samples, the multi-task learning model performs better than a similar multilayer perceptron single-task model. The difference is significant for the training samples of size 300 and 500. For the training sample of size 100, the difference is not statistically significant.

6.2. LAEI dataset

6.2.1. Multi-task learning

Table 5 shows the average R^2 scores of LUR models using different degrees of multi-task learning and single-task learning on the LAEI dataset. The best performing model for each training sample is in bold type.

Table 5. Average R^2 scores on the test samples from the LAEI dataset using multilayer perceptron

models with different numbers of shared layers.

	Shared layers			
Samples	0	1	2	Increase
100	0.489	**0.490**	0.476	+0.32%
300	**0.506**	0.468	0.490	−3.09%
500	0.514	**0.515**	0.507	+0.18%
3000	0.522	0.528	**0.534**	+2.25%

The results show an increase of the R^2 scores when using multi-task learning for models trained with 100, 500 and 3000 samples, while for 300 samples the single-task model performs better.

Similarly to the OpenSense dataset, the results show that there is no one-fits-all optimal structure of the model, with the optimal amount of shared layers varying with the training sample considered. A clear increase of the R^2 score with increasing degree of multi-task learning can however be seen when using a large training set of 3000 samples. The one-way ANOVA performed for each training set individually shows, that the modelling approaches differ significantly ($p < .001$).

6.2.2. Comparison

The comparison between the different models for the LAEI dataset can be seen in table 6.

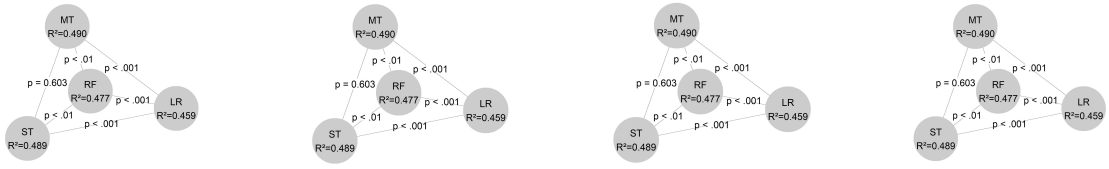
Table 6. Average R^2 scores on the test samples from the LAEI dataset using linear regression (LR), random forest regression (RF), as well as single-task learning (ST) and multi-task learning (MT) using a multilayer perceptron (MLP).

			MLP	
Samples	LR	RF	ST	MT
100	0.459	0.477	0.489	0.490
300	0.488	0.527	0.506	0.490
500	0.499	0.537	0.514	0.515
3000	0.505	0.572	0.522	0.534

Similarly to the OpenSense dataset, the models have been compared using pairwise Mann-Whitney U-tests. The results are shown in figure 4.

The comparison shows that the random forest model performs significantly better than other models and linear regression offers the statistically significant worse fit.

When comparing single-task and multi-task learning multilayer perceptron models, the results show an increased fit of the models using a multi-task learning approach compared to single-task models when trained using 100, 500 and 3000 samples. However, the difference is only statistically significant when trained with 3000 samples. When using 300 training samples, the multi-task learning model performs worse than the single-task learning model.



(a) 100 training samples (b) 300 training samples (c) 500 training samples (d) 3000 training samples

Figure 4. Pairwise Mann-Whitney U-tests between linear regression models (LR), random forest models (RF) and multilayer perceptron models using single-task (ST) and multi-task learning (MT) on the London Atmospheric Emissions Inventory dataset.

7. Discussion

The comparison of all the single-task models, including the baselines, shows a clear advantage of the random forest models over all other model types. This holds true for both datasets and all examined sample sizes. It is not an unexpected result, as previous comparisons have shown that random forest models provide high accuracies in the context of LUR [14, 42].

When comparing multi-task learning with the single-task learning approach on the multilayer

perceptron models the results for both the OpenSense and LAEI datasets indicate a possible increase in performance of the models when using a shared representation. However, the increase in performance is not large enough to surpass the random forest baseline model, which still outperforms the multi-task learning model.

In this section we discuss possible reasons for this limitation, what can be done to increase the benefits of multi-task learning and why it can still be a promising approach.

7.1. Task relatedness

Caruana [8] argues that multi-task learning helps improve generalization when using related tasks. Two tasks are defined to be related if they use the same variables to predict the outcome and if they use those variables in the same way [8].

Using this definition, it is possible to explore the relatedness of the tasks by comparing the relative feature importance between different pollutants. If two tasks (modelling two different pollutants) depend stronger on the same set of features and less so on different features, the tasks is considered highly related.

We used the permutation variable importance measure introduced by Fisher et al. [43] on the baseline random forest regression models to calculate the feature importance for all of the training samples. Figure 5 shows the feature importance calculated for the OpenSense dataset with 500 training samples and figure 6 for the LAEI dataset with 3000 samples. For all the other training samples the calculations show a very similar pattern of feature importance.

For the OpenSense dataset, the tasks appear to be less related, as the feature importance values vary strongly between the pollutants (figure 5). Figure 6 shows that all pollutants, except $PM_{2.5}$, depend similarly on the features. It is therefore reasonable to assume that the tasks of modelling different pollutants in this dataset are highly related.

The feature importance analysis does not paint a clear picture of how task relatedness translates into performance gain from shared representation. In our experiments, models on both datasets benefit from the multi-task approach even though the task relatedness, as measured by feature importance, is higher for the LAEI dataset.

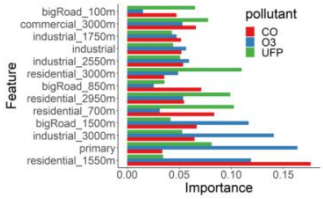


Figure 5. Feature importance for the OpenSense dataset with 500 samples.

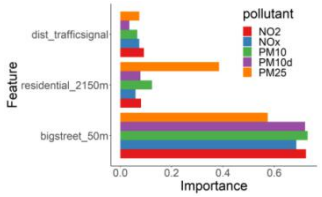


Figure 6. Feature importance for the London Atmospheric Emissions Inventory dataset with 3000.

7.2. Feature selection

One possible explanation for this unclear relationship between task relatedness and the advantage of multi-task learning could be the used feature selection procedure. As described in Section 3.3.1, variables used for training the models have been selected from a large pool of 244 features generated from OpenStreetMap. The selection procedure involved comparing the R^2 scores of linear models built using each of the features and including the best one.

We used an average over all pollutants to calculate the score for each feature. An alternative approach, which can be explored in further research, would be to select important features for each pollutant individually and then consider an aggregate of those features for the multi-task learning. However, because a shared metric was used only features that could on average predict all pollutant concentrations reasonably well were included in the pool of variables to be used for the multi-task learning models.

This selection procedure had two important consequences: First, it introduced a bias to the feature

importance metric as calculated in Section 7.1. All features that would offer very accurate models for only one of the pollutants, but not for the others, are not selected. Since only those features were selected that on average predicted concentrations for all pollutants reasonably well, the tasks are more related when comparing their feature importance than if a different feature selection method was used.

Second, selecting features that on average predict all pollutants well is in itself a form of multi-task learning. In fact, sparsity-enforcing regularization techniques have been used for linear models in the context of multi-task learning [22, 44]. Arguably feature selection is also one of the core mechanisms how multi-task learning improves prediction scores in multilayer perceptron models [8]. While the multi-task learning models considered for modelling pollutants still benefit from a shared hidden representation, the single-task models are not truly independent, as they all depend on features that have been selected using a multi-task method, possibly decreasing the observed difference.

7.3. Data quality

As discussed in Section 3.1.4 measurements for two of the pollutants within the OpenSense dataset are possibly noisy and only one pollutant offers high quality measurements. In contrast, the LAEI offers estimated concentrations of air pollutants which are not directly measured, but instead modelled using a atmospheric dispersion model.

As neither of the datasets offers high-quality data from physical monitoring stations of air pollutant concentrations, the question arises of how well the findings would generalize to such a hypothetical dataset. While a definitive answer can only be given by examining multi-task learning on such a dataset, there are some arguments that can be made on why our approach would still work.

As can be seen on the OpenSense dataset, multi-task learning increases the fit of the model compared to a similar single-task learning model for all pollutants, including ultrafine particles for which high-quality measurements are available. Thus, since including noisy measurements can improve the prediction accuracy of high-quality data when modelling in a multi-task learning context, it is reasonable to believe that a similar effect would be observed if high-quality data was used as the additional tasks.

The LAEI dataset, on the other hand, offers only modelled concentrations. While air dispersion models will always offer a simplified model of the emissions and spread of air pollution, they generate accurate general trends, especially when only long averaging periods of one year or more are considered. Thus, a similar benefit of multi-task learning can be expected when accurately measured data is used.

Both single- and multi-task models are trained on equal data quality. The results show that multi-task learning models offer better prediction performance than similar multilayer perceptron single-task models. It is, however, unclear how the difference would manifest when comparing the predictions from models trained on accurate air pollution data from high-end monitoring stations. Especially when the sources of error are not independent, the multi-task models might only learn the noise patterns in the data. While the findings on the OpenSense data indicate this not to be the case, additional experiments using poor-quality data with independent sources of error could rule out this possibility.

7.4. Sample size

It is a known observation in machine learning that small sample sizes often lead to overfitting, especially when using complex models like neural networks as compared to traditional models (e.g. linear regression) [45, 46]. This limitation makes applying complex models in the context of LUR difficult, since high-quality measurements are often a limited resource as mentioned in Section 1.

Our results clearly confirm this pattern, as more training samples lead to better prediction scores for both datasets and all considered model types. When comparing multi-task and single-task learning models, the advantage of a shared representation approach only becomes apparent with sufficient training data. For the LAEI dataset the largest positive effect appears for 3000 training samples, while with less data single-task models do not differ significantly from multi-task models or perform even better. Multi-task learning shows a significant advantage for the 300 and 500 samples subset of the

OpenSense dataset.

Large data requirements make the application of multitask learning models for LUR difficult, as datasets containing air pollutant concentrations usually contain limited numbers of samples.

8. Conclusions and future work

In this work, we assessed multi-task learning for LUR. As pollutants are often monitored together, the potential dependence on the same set of factors makes modelling several pollutants simultaneously an attractive possibility. The results do indeed show that multi-task learning models perform significantly better than similar multilayer perceptron single-task learning models when using a large enough training set.

However, for both datasets that have been considered -the London Atmospheric Emissions Inventory and the OpenSense dataset, random forest regression still outperforms the multi-task learning models for all training samples. A possible direction for future research is the application of multitask learning using tree-based ensemble methods [47]. Nonparametric ensemble models might overcome the large data requirements of multilayer perceptron models while still benefiting from shared information between different pollutants.

In order to decrease the data requirements for multilayer perceptron models it might be worthwhile to explore pretraining with weak labels. Interpolation methods may be used to produce dense maps of pollution estimates from measurements which can be used as weak labels. These labels can be used to train the model. Thereafter, the model can be fine-tuned using only labels from real measurements. This training procedure might improve multi-layer perceptron model performance for LUR, where there are often relatively few data points.

Another promising research direction is the application of multi-task learning for deep-learning based LUR models like MapLUR [18]. This model has shown better performance than random forests on the dataset of the London Atmospheric Emissions Inventory for single-task learning and our results suggest that multi-task learning can further increase performance.

Future work should also explore different feature selection methods, as more liberal selection procedures might allow for higher variability in feature dependence between different pollutants and consequently multi-task learning might benefit even more from a shared representation. Especially sparsity-enforcing regularization techniques used for multivariate linear regression [22, 44] might be a promising approach to building LUR models using multi-task learning.

As high-quality air pollution datasets mostly contain only a limited number of measurement locations, the experiments have been performed on data obtained from low-cost sensors in the case of the OpenSense dataset and modelled air concentrations using an atmospheric dispersion model in the case of the London Atmospheric Emissions Inventory. An important direction for future research would be to compare multi-task learning and single-task learning on a large-scale dataset containing high-quality measurements.

Overall, the multi-task learning method using multilayer perceptrons shows better performance than similar single-task models, while still being outperformed by Random Forest models. However, this work demonstrates the potential of learning shared representations for better air pollution prediction performance, which can be explored in further research using different model types.

References

[1] R. Beelen *et al*., “Effects of long-term exposure to air pollution on natural-cause mortality: an analysis of 22 European cohorts within the multicentre ESCAPE project,” *Lancet*, vol. 383, no. 9919, pp. 785–795, Mar 2014.

[2] M. Bauer *et al*., “Urban particulate matter air pollution is associated with subclinical atherosclerosis: results from the HNR (Heinz Nixdorf Recall) study,” *J. Am. Coll. Cardiol.*, vol. 56, no. 22, pp. 1803–1808, Nov 2010.

[3] A. P. C. Takano *et al*., “Pleural anthracosis as an indicator of lifetime exposure to urban air pollution: An autopsy-based study in Sao Paulo,” *Environ. Res.*, vol. 173, pp. 23–32, 06 2019.

[4] E. E. Agency, *Air Quality in Europe - 2019 Report*. Luxembourg: Publications office of the European Union, 2019.
[5] P. Hystad *et al*., "Creating national air pollution models for population exposure assessment in canada," *Environmental health perspectives*, vol. 119, no. 8, pp. 1123– 1129, Aug 2011.
[6] BAFU and EMPA, "Technischer bericht zum nationalen beobachtungsnetz für luftfremdstoffe (nabel)," 2018.
[7] P. H. Ryan *et al*., "A comparison of proximity and land use regression traffic exposure models and wheezing in infants," *Environmental Health Perspectives*, vol. 115, no. 2, pp. 278–284, 2007.
[8] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.
[9] Y.-H. Dai and W.-X. Zhou, "Temporal and spatial correlation patterns of air pollutants in chinese cities," *PLOS ONE*, vol. 12, no. 8, pp. 1–24, 08 2017.
[10] P. H. Ryan and G. K. LeMasters, "A review of land-use regression models for characterizing intraurban air pollution exposure,"*Inhalation toxicology*, vol. 19 Suppl 1, no. Suppl 1, pp. 127–133, 2007.
[11] H. Amini *et al*.,"A systematic review of land use regression models for volatile organic compounds," *Atmospheric Environment*, vol. 171, pp. 1 – 16, 2017.
[12] L. Li, J. Wu *et al*.,"Use of generalized additive models and cokriging of spatial residuals to improve land-use regression estimates of nitrogen oxides in southern california," *Atmospheric Environment*, vol. 55, pp. 220 – 228, 2012.
[13] B. Zou *et al*.,"Satellite based mapping of ground pm2.5 concentration using generalized additive modeling,"*Remote Sensing*, vol. 9, no. 1, p. 1, Dec 2016.
[14] C. Brokamp *et al*., "Exposure assessment models for elemental components of particulate matter in an urban environment: A comparison of regression and random forest approaches,"*Atmospheric Environment*, vol. 151, pp. 1 – 11, 2017.
[15] S. Araki *et al*., "Spatiotemporal land use random forest model for estimating metropolitan no2 exposure in japan," *Science of The Total Environment*, vol. 634, pp. 1269 – 1277, 2018.
[16] M. S. Alam and A. McNabola,"Exploring the modeling of spatiotemporal variations in ambient air pollution within the land use regression framework: Estimation of pm10 concentrations on a daily basis,"*Journal of the Air & Waste Management Association*, vol. 65, no. 5, pp. 628–640, 2015.
[17] M. D. Adams and P. S. Kanaroglou, "Mapping realtime air pollution health risk for environmental management: Combining mobile and stationary air pollution monitoring with neural network models," *Journal of Environmental Management*, vol. 168, pp. 133 – 141, 2016.
[18] M. Steininger *et al*., "Maplur: Exploring a new paradigm for estimating air pollution using deep learning on map images," 2020.
[19] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org," https://www.openstreetmap.org, 2017.
[20] F. Lautenschlager *et al*., "Openlur: Off-the-shelf air pollution modeling with open features and machine learning,"*Atmospheric Environment*, vol. 233, p. 117535, 2020.
[21] H. Guo *et al*., "Characterization of criteria air pollutants in beijing during 2014–2015," *Environmental Research*, vol. 154, pp. 334 – 344, 2017.
[22] S. Ruder, "An overview of multi-task learning in deep neural networks," *CoRR*, vol. abs/1706.05098, 2017.
[23] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 160–167.
[24] X. Gibert *et al*.,"Deep multi-task learning for railway track inspection,"*IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 153–164, Jan 2017.

[25] B. Ramsundar *et al*.,"Massively multitask networks for drug discovery," 2015.
[26] K. Aberer *et al*.,"Opensense: Open community driven sensing of environment," in *Proceedings of the ACM SIGSPATIAL International Workshop on GeoStreaming, ser*. IWGS '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 39–42.
[27] A. Q. T. G. L. Authority).,"London atmospheric emissions inventory (laei)." https://data.london.gov.uk/dataset/london-atmospheric-emissions-inventory-2013, 2013.
[28] J. J. Li et al., "Sensing the air we breathe: The opensense zurich dataset," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, ser. AAAI'12. AAAI Press, 2012, p. 323–325.
[29] D. Hasenfratz *et al*., "Pushing the spatio-temporal resolution limit of urban air pollution maps," in *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, March 2014, pp. 69–77.
[30] M. Johnson et al., "Development of temporally refined land-use regression models predicting daily household-level air pollution in a panel study of lung function among asthmatic children," *Journal of Exposure Science & Environmental Epidemiology*, vol. 23, no. 3, pp. 259–267, 2013.
[31] D. Roberts–Semple *et al*., "Seasonal characteristics of ambient nitrogen oxides and ground–level ozone in metropolitan northeastern new jersey," *Atmospheric Pollution Research*, vol. 3, no. 2, pp. 247 – 257, 2012.
[32] R. D. Peng *et al*.,"Seasonal Analyses of Air Pollution and Mortality in 100 US Cities,"*American Journal of Epidemiology*, vol. 161, no. 6, pp. 585–594, 03 2005.
[33] Council of European Union, "Directive 2008/50/ec of the european parliament and of the council," http://data.europa.eu/eli/dir/2008/50/2015-09-18, 2008.
[34] D. Hasenfratz *et al*., "On-the-fly calibration of low-cost gas sensors," in *Wireless Sensor Networks*, G. P. Picco and W. Heinzelman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 228– 244.
[35] W. R. Ott, "A physical explanation of the lognormality of pollutant concentrations," *Journal of the Air & Waste Management Association*, vol. 40, no. 10, pp. 1378–1383, 1990.
[36] B. Maag *et al*.,"Ultrafine particle dataset collected by the opensense zurich mobile sensor network," https://zenodo.org/record/3298842, Sep 2018.
[37] Maag, Balz et al., "Ozone and carbon monoxide dataset collected by the opensense zurich mobile sensor network," https://zenodo.org/record/3355208, Jul 2019.
[38] M. L. Bermingham et al., "Application of high-dimensional feature selection: evaluation for genomic prediction in man,"*Scientific reports*, vol. 5, pp. 10312– 10312, May 2015.
[39] J. Baxter, "A bayesian/information theoretic model of learning to learn via multiple task sampling," *Machine Learning*, vol. 28, no. 1, pp. 7–39, Jul 1997.
[40] F. Chollet et al., "Keras," https://keras.io, 2015.
[41] G. Hoek *et al*., "A review of land-use regression models to assess spatial variation of outdoor air pollution," *Atmospheric Environment*, vol. 42, no. 33, pp. 7561 – 7578, 2008.
[42] X. Hu *et al*., "Estimating pm2.5 concentrations in the conterminous united states using the random forest approach," *Environmental Science & Technology*, vol. 51, no. 12, pp. 6936–6944, 2017.
[43] A. Fisher *et al*., "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously," 2018.
[44] A. Argyriou, *et al*., "Convex multi-task feature learning," *Machine Learning*, vol. 73, no. 3, pp. 243–272, Dec 2008.
[45] A. Vabalas *et al*., "Machine learning algorithm validation with a limited sample size," *PloS one*, vol. 14, no. 11, pp. e0224365– e0224365, Nov 2019.
[46] S. J. Raudys and A. K. Jain, "Small sample size effects in statistical pattern recognition: recommendations for practitioners," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 252–264, 1991.

[47] J. Simm *et al*., "Tree-based ensemble multi-task learning method for classification and regression," *IEICE Transactions on Information and Systems*, vol. E97.D, no. 6, pp. 1677–1681, 2014.

Deep Learning for Climate Model Output Statistics

Michael Steininger[1], **Daniel Abel**[2], **Katrin Ziegler**[2], **Anna Krause**[1], **Heiko Paeth**[2], **and Andreas Hotho**[1]

[1]Chair of Computer Science X (Data Science), University of Würzburg
{steininger, anna.krause, hotho}@informatik.uni-wuerzburg.de
[2]Chair of Physical Geography, University of Würzburg
{daniel.abel, katrin.ziegler, heiko.paeth}@uni-wuerzburg.de

Abstract

Climate models are an important tool for the assessment of prospective climate change effects but they suffer from systematic and representation errors, especially for precipitation. Model output statistics (MOS) reduce these errors by fitting the model output to observational data with machine learning. In this work, we explore the feasibility and potential of deep learning with convolutional neural networks (CNNs) for MOS. We propose the CNN architecture ConvMOS specifically designed for reducing errors in climate model outputs and apply it to the climate model REMO. Our results show a considerable reduction of errors and mostly improved performance compared to three commonly used MOS approaches.

1 Introduction

An important source of information for the prospective effects of climate change are numerical climate models such as general circulation models (GCMs) and regional climate models (RCMs). However, these climate models often exhibit systematic errors and deficiencies in representations of climate processes which limit the quality of the resulting projections. This problem is especially pronounced for precipitation. It is therefore common to apply model output statistics (MOS), which are statistical post-processing techniques to reduce these errors. MOS approaches correct the modeled precipitation to correspond more closely to observational data. This allows us to study future climate conditions and the effects of climate change more accurately especially at a local scale [1].

There are two general approaches to MOS – distribution-wise MOS and event-wise MOS. Distribution-wise MOS corrects the distribution of the simulated variable by mapping means and other distribution characteristics to the observational distribution. Event-wise MOS directly links the simulated and observed time series through statistical models, which generally performs better than distribution-wise MOS [2]. We therefore consider event-wise MOS in this work.

A number of approaches to event-wise MOS have been used in previous work. A very simple approach is local scaling where an individual Linear Regression is fitted per location of interest, which has shown to work reasonably well [2]. Other works propose non-local MOS approaches, where for each location the MOS is aware of the climatic conditions at nearby locations. This can lead to a large number of predictors for the MOS, which is why dimensionality reduction techniques, e. g. principal component analysis (PCA), are often applied [1, 2, 3, 4]. Non-local MOS has been done with a range of machine learning models namely Linear Regression [1, 2], Random Forests (RFs) [3, 4], Support Vector Machines (SVMs) [3, 5, 6], and Multilayer Perceptrons (MLPs) [7].

While these methods have proven to be effective, we believe that there is considerable potential in exploring the use of advanced deep learning techniques. Especially convolutional neural networks (CNNs) [8] have shown proficiency in tasks with geospatial data [9, 10], which indicates potential for

Tackling Climate Change with Machine Learning workshop at NeurIPS 2020.

novel non-local MOS with this type of neural network. We believe that their ability to learn spatial patterns is well suited for reducing systematic errors in climate models. It is therefore promising to assess how this can be used for MOS and whether there is potential for performance improvements.

In this work, we examine the feasibility and potential of convolutional deep learning models as MOS. Thus, we propose the CNN architecture ConvMOS specifically designed for climate MOS and apply it to correcting simulated precipitation of the RCM REMO [11, 12, 13]. Our results show that ConvMOS can reduce errors considerably, providing mostly better performance than three commonly used MOS approaches. This suggests that our proposed approach is feasible and promising.

2 Dataset

Model Data For our study we use daily data of the hydrostatic version of the RCM REMO (version REMO2015) [11, 12, 13] for the period 2000 to 2015. Our study area has been defined over an extended German region with $0.11° \times 0.11°$ resolution covering the area from $-1.43°$ to $22.22°$ E and $42.77°$ to $57.06°$ N (GER-11). We use the following 22 MOS predictors from REMO: Daily mean, minimum and maximum temperature $2\,\mathrm{m}$ above surface [K], u- and v-wind $10\,\mathrm{m}$ above surface $[\mathrm{m\,s^{-1}}]$, sea level pressure [Pa] and total precipitation (convective + large scale + snowfall) [mm]. Further, the temperature [K], geopotential height [m], and specific humidity $[\mathrm{kg\,kg^{-1}}]$ in the pressure levels of 100, 200, 500, 850 and 950 hPa are used. For elevation information the dataset GTOPO ($0.009° \times 0.009°$) [14, 15] is used by REMO, which we also employ as another predictor for ConvMOS. More specifics about the climate model can be found in Appendix A.

Observational Data For observational data we use the gridded dataset E-OBS [16] version 19.0e which is based on an ensemble of interpolated station data [17]. Since the station density varies in space and time, the interpolation of the station data has some uncertainty [17]. Amongst other variables E-OBS provides daily precipitation sums at a $0.1°$ resolution, which is our predictand. The grids of the model and observational data are interpolated bilinearly to the same $0.11°$ grid [18].

3 Deep Learning for Climate Model Output Statistics

To explore the use of deep learning and CNNs as MOS we propose the architecture ConvMOS.

Idea The basic idea of ConvMOS stems from two potential sources of error in climate models: First, specific location errors which typically stem from poor grid point representation of topography [1, 2]. Second, systematic errors originating from parameterization, which replaces too complex or too small-scale processes with simplified variants. For precipitation, cloud and rainfall formation is based on parameterization, leading to an overestimation over land [1].

To efficiently reduce both types of errors we combine per-location model parameters, which can learn the characteristics of a specific location, and global model parameters, which can learn spatial precipitation patterns to efficiently help reduce systematic errors in climate models. Thus, we define two module types: Local network and global network.

Local Network The local network module contains individual model parameters for each location in the study area, allowing it to reduce specific local errors. It is implemented with a linearly activated 1D CNN where the input at each time is first reshaped so that it has the dimensions $(\text{height} * \text{width}, \text{predictors})$ instead of $(\text{predictors}, \text{height}, \text{width})$. In conjunction with setting the kernel size equal to the number of predictors, this allows us to group the convolution for each input channel (i.e. each location) so that each location is convolved with its own set of filters for all predictors. Thus each location has its own model parameters, in which location characteristics can be encoded. This module is not provided with elevation data as it would be static across all times for each location. The output of the local network is a grid with precipitation residuals for each location.

Global Network The global network learns spatial patterns in precipitation and other predictors. This can be done efficiently with CNNs [19]. The module contains a 2D CNN with 4 layers which learns useful filters for the reduction of systematic errors across the study area using all predictors. Starting from the first layer, the layers have 4, 8, 16, and 1 filters and kernel sizes of 9, 1, 5, and 3

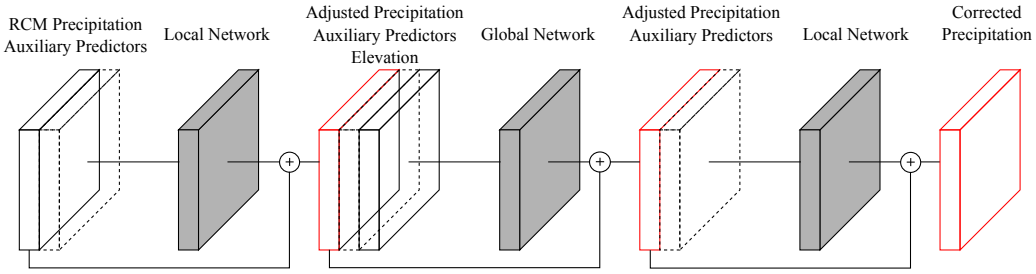


Figure 1: Architecture of ConvMOS.

respectively. Each convolutional layer has its padding parameter set to half its kernel size (rounded down to the nearest whole number) which leads to the output of each layer having the same width and height as its input. All layers use the ReLU [20] activation function, a stride of 1, and a dilation of 1. As with the local network, this module also outputs a grid of precipitation residuals.

Architecture The architecture is depicted in Figure 1. It expects a 3D input with dimensions $(\text{predictors}, \text{height}, \text{width})$ for each time step. The data is sequentially passed through three modules (depicted in gray) where each module adjusts the precipitation input with the goal of reducing the error. The architecture employs so called "shortcut connections" for each module where each module's output is added to its precipitation input, which eases training for neural networks [21]. We first apply a local network to correct any specific local errors. Thereafter, the global network aims to reduce any remaining systematic errors across the study area. Finally, another local network makes sure that the systematic corrections of the global network are not introducing new local errors. The training procedure is described in Appendix B.

4 Experiment

To evaluate ConvMOS we apply it to the data described in Section 2. We also apply three other commonly used MOS approaches, a local Linear Regression, a non-local Principal Component Regression approach and a non-local RF method, for comparison.

Local Linear Regression (Lin) For each cell in the study area a separate Linear Regression is fitted where the predictor is the simulated precipitation and the predictand is the observed precipitation. This approach is local in that each Linear Regression is unaware of conditions in nearby cells [2].

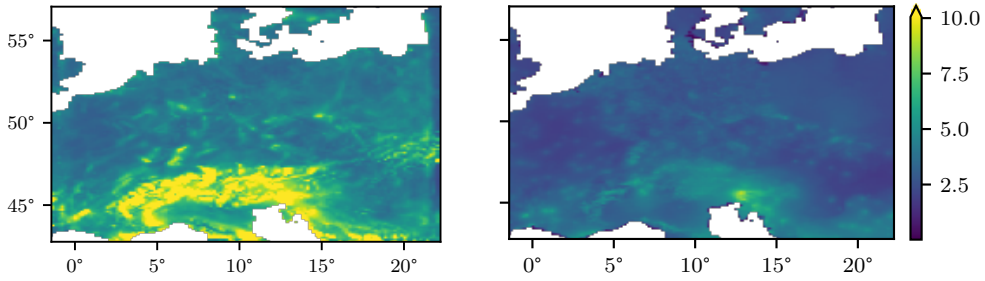
Non-local Principal Component Regression (NL PCR) Instead of only using the large-scale conditions at a specific location for a Linear Regression, we provide all available predictors at each nearby location which is at most ± 5 cells away in either direction on the grid. To reduce the dimensionality of the predictors, we apply a supervised PCA [22], which is explained in Appendix D [2].

Non-local Random Forest (NL RF) For the non-local Random Forest MOS approach we provide all available predictors of each location ± 5 cells away, as with the non-local PC regression approach. Following [3] and [4] we also apply a supervised PCA (see Appendix D). Each location is fitted with its own RF. Hyperparameters are optimized at each location individually (see Appendix C).

Experimental Setup We split the 16 years of daily data into a training (2000–2009), a validation (2010), and a test set (2011–2015). All predictors are standardized based on the training set so that they have a mean of zero and a standard deviation of one. We tried different hyperparameters for our architecture and selected the ones presented in this work based on the validation set performance. All results reported in the following are based on the unseen test set after fitting the MOS on the training set. For evaluation we use a number of common MOS metrics, namely RMSE, Pearson Correlation, Skill Score [23], R^2, and bias to assess different aspects of MOS performance. ConvMOS is trained 10 times since its fitting method is, in contrast to the linear comparison methods, non-deterministic. This results in slight performance differences for each fitted instance. Despite its inherent randomness, the RF method is only calculated once since this computation already took over four days for our study area with 15 CPU cores in parallel.

Table 1: Experimental results. Mean metrics on the test set for all study area locations available in observational data. All means and standard deviations are rounded to two decimal places. Correlation mean is calculated with Fisher's z-transformation [24].

MOS \ Metric	RMSE	Corr.	Skill	R^2	Bias
None	5.32	0.49	**0.93**	−28.24	0.31
Lin	3.77	0.49	**0.93**	0.23	0.03
NL PCR	3.37	0.62	0.92	0.36	**0.02**
NL RF	3.39	0.61	0.81	0.36	0.03
ConvMOS	**2.99** ± **0.01**	**0.72** ± **0.00**	0.92 ± 0.00	**0.49** ± **0.01**	−0.10 ± 0.06



(a) REMO raw

(b) ConvMOS

Figure 2: RMSE of precipitation in mm for the test set across the study area. Note that there are some cells in REMO raw with considerably higher RMSE than $10\,\mathrm{mm}$ but we limited the colorbar's extent for better visibility of the general performance.

Results Table 1 shows the mean metrics on the test set for all study area locations available in observational data (i.e. land points). All MOS approaches improve all metrics considerably when compared to applying no MOS, except for the skill score. This means that the precipitation distribution of REMO is already rather close to that of the observations with a skill score of 0.93 and can barely be improved by the MOS methods. ConvMOS is showing the best performance of all tested MOS approaches for the metrics RMSE, correlation, and R^2. This indicates that our approach is able to estimate precipitation more accurately than all considered comparison methods. The skill score is very close but still reduced slightly by 0.01 compared to the best value. ConvMOS shows less bias than REMO but it seems to have a tendency to underestimate precipitation. The other approaches to tend to overestimate, but to a lesser extent. ConvMOS is also showing rather stable performance as can be seen on the standard deviations in Table 1 despite its non-deterministic fitting procedure. We also ran this experiment with precipitation as the only climate predictor as some prior work has done [2, 3, 4, 6] but found all methods to perform worse without additional predictors.

Figure 2 visualizes RMSEs for all locations with observational data across the study area for the raw REMO output and ConvMOS. We can see that our approach reduces error across most locations. Especially the precipitation in the Alps and other mountainous regions is improved considerably.

5 Conclusion

In this work we explored the feasibility and possibilities of deep learning MOS. To this end, we proposed the CNN-based ConvMOS architecture specifically designed to reduce errors in climate model outputs, which we applied to the RCM REMO. All in all, the initial results for ConvMOS seem promising. Our MOS approach is able to improve the daily precipitation data considerably. Improvements in MOS allow for more accurate climate data especially at high spatial resolutions. While our approach mostly provides better performance than the other standard approaches considered here, additional comparisons have to be made in the future with other MOS techniques and data from different climate models.

Acknowledgments and Disclosure of Funding

This research was conducted in the BigData@Geo project supported by the European Regional Development Fund (ERDF).

References

[1] Heiko Paeth. Postprocessing of simulated precipitation for impact research in west africa. part i: model output statistics for monthly data. *Climate Dynamics*, 36(7-8):1321–1336, 2011.

[2] Jonathan M Eden and Martin Widmann. Downscaling of gcm-simulated precipitation using model output statistics. *Journal of Climate*, 27(1):312–324, 2014.

[3] Zulfaqar Sa'adi, Shamsuddin Shahid, Eun-Sung Chung, and Tarmizi bin Ismail. Projection of spatial and temporal changes of rainfall in sarawak of borneo island using statistical downscaling of cmip5 models. *Atmospheric research*, 197:446–460, 2017.

[4] Muhammad Noor, Tarmizi bin Ismail, Shahid Ullah, Zafar Iqbal, Nadeem Nawaz, and Kamal Ahmed. A non-local model output statistics approach for the downscaling of cmip5 gcms for the projection of rainfall in peninsular malaysia. *Journal of Water and Climate Change*, 2019.

[5] Sahar Hadi Pour, Shamsuddin Shahid, Eun-Sung Chung, and Xiao-Jun Wang. Model output statistics downscaling using support vector machine for the projection of spatial and temporal changes in rainfall of bangladesh. *Atmospheric Research*, 213:149–162, 2018.

[6] Kamal Ahmed, Shamsuddin Shahid, Nadeem Nawaz, and Najeebullah Khan. Modeling climate change impacts on precipitation in arid regions of pakistan: a non-local model output statistics downscaling approach. *Theoretical and Applied Climatology*, 137(1-2):1347–1364, 2019.

[7] Sanaz Moghim and Rafael L Bras. Bias correction of climate modeled temperature and precipitation using artificial neural networks. *Journal of Hydrometeorology*, 18(7):1867–1884, 2017.

[8] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[9] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. In *Advances in neural information processing systems*, pages 5617–5627, 2017.

[10] Michael Steininger, Konstantin Kobs, Albin Zehe, Florian Lautenschlager, Martin Becker, and Andreas Hotho. Maplur: Exploring a new paradigm for estimating air pollution using deep learning on map images. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 6(3):1–24, 2020.

[11] Detlev Majewski. The europa-modell of the deutscher wetterdienst. *ECMWF Proc." Numerical Methods in atmospheric models." Reading*, 2:147–191, 1991.

[12] Daniela Jacob. A note to the simulation of the annual and inter-annual variability of the water budget over the baltic sea drainage basin. *Meteorology and Atmospheric Physics*, 77(1-4):61–73, 2001.

[13] D Jacob, BJJM Van den Hurk, Ulf Andrae, G Elgered, C Fortelius, LP Graham, SD Jackson, U Karstens, Chr Köpken, R Lindau, et al. A comprehensive model inter-comparison study investigating the water budget during the baltex-pidcap period. *Meteorology and Atmospheric Physics*, 77(1-4):19–43, 2001.

[14] EDC DAAC. Gtopo 30 database. eros data center distributed active archive center. *US Geological Survey, EROS Data Center, Sioux Falls, South Dakota*, 1996.

[15] Dean B Gesch, Kristine L Verdin, and Susan K Greenlee. New land surface digital elevation model covers the earth. *Eos*, 80(6):69–70, 1999.

[16] M R Haylock, N Hofstra, A M G Klein Tank, E J Klok, P D Jones, and M New. A European daily high-resolution gridded data set of surface temperature and precipitation for 1950-2006. *Journal of Geophysical Research Atmospheres*, 113(20):D20119, 2008.

[17] Richard C Cornes, Gerard van der Schrier, Else JM van den Besselaar, and Philip D Jones. An ensemble version of the e-obs temperature and precipitation data sets. *Journal of Geophysical Research: Atmospheres*, 123(17):9391–9409, 2018.

[18] Uwe Schulzweida. CDO User Guide, 2019.

[19] Thomas Vandal, Evan Kodra, Sangram Ganguly, Andrew Michaelis, Ramakrishna Nemani, and Auroop R Ganguly. Deepsd: Generating high resolution climate change projections through single image super-resolution. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, pages 1663–1672, 2017.

[20] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Eric Bair, Trevor Hastie, Debashis Paul, and Robert Tibshirani. Prediction by supervised principal components. *Journal of the American Statistical Association*, 101(473):119–137, 2006.

[23] SE Perkins, AJ Pitman, NJ Holbrook, and J McAneney. Evaluation of the ar4 climate models' simulated daily maximum temperature, minimum temperature, and precipitation over australia using probability density functions. *Journal of climate*, 20(17):4356–4376, 2007.

[24] N Clayton Silver and William P Dunlap. Averaging correlation coefficients: Should fisher's z transformation be used? *Journal of Applied Psychology*, 72(1):146, 1987.

[25] Erich Roeckner, Klaus Arpe, Lennart Bengtsson, M. Christoph, Martin Claussen, Lydia Dümenil, Monika Esch, Marco Giorgetta, Ulrich Schlese, and Uwe Schulzweida. The atmospheric general circulation model ECHAM-4: Model description and simulation of present-day climate. Technical report, Max-Planck-Institute of Meteorology, Hamburg, 1996.

[26] Stefan Hagemann. An Improved Land Surface Parameter Dataset for Global and Regional Climate Models. Technical Report 336, Max Planck Institute for Meteorology, Hamburg, 2002.

[27] Tido Semmler. *Der Wasser- und Energiehaushalt der arktischen Atmosphäre*. PhD thesis, Hamburg, 2002.

[28] Sven Kotlarski. *A Subgrid Glacier Parameterisation for Use in Regional Climate Modelling*. PhD thesis, Hamburg, 2007.

[29] Joni-Pekka Pietikäinen. The regional aerosol-climate model REMO-HAM. *Geoscientific Model Development*, 5:1323–1339, 2012.

[30] D Dee, S Uppala, A Simmons, P Berrisford, P Poli, S Kobayashi, U Andrae, M A Balmaseda, G Balsamo, P Bauer, P Bechtold, A Beljaars, L van den Berg, J Bidlot, N Bormann, C Delsol, R Dragani, M Fuentes, A J Geer, L Haimberger, S B Healy, H Hersbach, E V Holm, L Isaksen, P Kallberg, M Köhler, M Matricardi, A P McNally, B M Monge-Sanz, J.-J. Morcrette, B.-K. Park, C Peubey, P de Rosnay, C Tavolato, J.-N. Thépaut, and F Vitart. The ERA-Interim reanalysis: configuration and performance of the data assimilation system. *Quarterly Journal of the Royal Meteorological Society*, 137:553–597, 2011.

[31] Paul Berrisford, D Dee, K Fielding, M Fuentes, P Kallberg, S Kobayashi, and S Uppala. The ERA-Interim Archive. Version 2.0. Technical report, ECMWF, Reading, 2011.

[32] Claas Teichmann. *Climate and Air Pollution Modelling in South America with Focus on Megacities*. Phd thesis, Hamburg, 2010.

[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[34] Rich Caruana, Steve Lawrence, and C Lee Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems*, pages 402–408, 2001.

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

Table 2: Search space for the RF hyperparameter random search.

Hyperparameter	Search space
n_estimators	10 – 2000
max_features	0.01 – 1.0
max_depth	10 – 110
min_samples_split	2 – 10
min_samples_leaf	1 – 10
bootstrap	True or False

A Climate Model Data

For our study we use daily data of the hydrostatic version of the RCM REMO (version REMO2015) [11, 12, 13] for the period 2000 to 2015. REMO is based on the Europa Modell [11] with the model of the GCM ECHAM4 [25] with some improvements implemented since then (e.g. [26, 27, 28, 29]). The reanalysis ERA-Interim ($0.75° \times 0.75°$) [30, 31] is used as forcing data, providing the lateral boundary conditions. The atmosphere's vertical resolution is represented by 27 hybrid levels with increasing distance to the top of the atmosphere. In lower levels they follow the topography [32]. As mentioned in the main paper the dataset GTOPO ($0.009° \times 0.009°$) [14, 15] is used by REMO for elevation information. Both model and observational data for the MOS methods is provided at $0.11°$ resolution. The data is arranged on a 2D grid with 121×121 cells or locations.

B ConvMOS Training Details

The architecture is fitted with the Adam optimizer [33], the mean squared error (MSE) as the loss function and a learning rate of 0.001. Only errors at locations where observational data is available were incorporated for the MSE. Training is conducted for at most $100\,000$ epochs. Early stopping is used to stop training when the validation MSE is not improving for more than 40 epochs in a row, preventing considerable overfitting [34].

C Random Forest Hyperparameter Optimization

Each location in our study area has its own RF instance for MOS which uses the RandomForestRegressor from scikit-learn [35]. Since RF performance depends considerably on its hyperparameters we look for optimal values with a random search. For each cell we train 20 RF instances on the training set with hyperparameter values sampled randomly from the search space shown in Table 2. Each instance is evaluated on the validation set. The RF instance with the best R^2 is then applied on the test set.

D Supervised Principal Component Analysis

Like other previous MOS approaches [1, 3, 4] we preprocess our predictors for the standard MOS methods to reduce dimensionality and remove potentially unhelpful information. Like [3] and [4] we use supervised PCA [22]. First, we select the best predictors based on a univariate regression. How many of the predictors are retained is set according to a grid search with our validation data. In this search we try all values between only choosing the single best predictor and using the 30 best predictors. Then, PCA reduces the dimensionality of these predictors, keeping the first components that explain $95\,\%$ of the variance [3].

Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-022-00877-6

ConvMOS: climate model output statistics with deep learning

Michael Steininger[1] · Daniel Abel[2] · Katrin Ziegler[2] · Anna Krause[1] · Heiko Paeth[2] · Andreas Hotho[1]

Received: 9 December 2021 / Accepted: 5 October 2022

Abstract

Climate models are the tool of choice for scientists researching climate change. Like all models they suffer from errors, particularly systematic and location-specific representation errors. One way to reduce these errors is model output statistics (MOS) where the model output is fitted to observational data with machine learning. In this work, we assess the use of convolutional Deep Learning climate MOS approaches and present the ConvMOS architecture which is specifically designed based on the observation that there are systematic and location-specific errors in the precipitation estimates of climate models. We apply ConvMOS models to the simulated precipitation of the regional climate model REMO, showing that a combination of per-location model parameters for reducing location-specific errors and global model parameters for reducing systematic errors is indeed beneficial for MOS performance. We find that ConvMOS models can reduce errors considerably and perform significantly better than three commonly used MOS approaches and plain ResNet and U-Net mod-

Responsible editor: Albrecht Zimmermann.

Michael Steininger
steininger@informatik.uni-wuerzburg.de

Daniel Abel
daniel.abel@uni-wuerzburg.de

Katrin Ziegler
katrin.ziegler@uni-wuerzburg.de

Anna Krause
anna.krause@informatik.uni-wuerzburg.de

Heiko Paeth
heiko.paeth@uni-wuerzburg.de

Andreas Hotho
hotho@informatik.uni-wuerzburg.de

[1] Chair of Computer Science X (Data Science), University of Würzburg, Würzburg, Germany

[2] Chair of Physical Geography, University of Würzburg, Würzburg, Germany

Published online: 27 October 2022

els in most cases. Our results show that non-linear MOS models underestimate the number of extreme precipitation events, which we alleviate by training models specialized towards extreme precipitation events with the imbalanced regression method DenseLoss. While we consider climate MOS, we argue that aspects of ConvMOS may also be beneficial in other domains with geospatial data, such as air pollution modeling or weather forecasts.

Keywords Neural networks · Climate · Model output statistics

1 Introduction

An important source of information for the prospective effects of climate change are numerical climate models such as general circulation models (GCMs) and regional climate models (RCMs). However, these models often exhibit systematic errors and deficiencies in representations of climate processes which limit the quality of the resulting projections. Especially the hydrological cycle is subject to uncertainty, amplifying this problem for precipitation. It is therefore common to apply model output statistics (MOS), which are statistical post-processing techniques to reduce these errors. MOS correct the modeled precipitation to correspond more closely to observational data. With climate change becoming a more and more severe issue, we believe that it is important for the data mining community to contribute to the global effort towards assessing and combating climate change by further improving MOS performance both in the mean and for extreme events. Better MOS allows us to study future climate conditions and effects of climate change more accurately (Paeth 2011).

Currently used climate MOS approaches typically rely on standard methods from statistics and machine learning like Linear Regression (Paeth 2011) and Random Forests (RFs) (Noor et al. 2019). For each location of interest a separate model instance is fitted to reduce errors in precipitation. These models are either local when they use large-scale atmospheric conditions at that specific location or non-local, when they also consider conditions at locations nearby.

In this work, we aim to further bridge the gap between climate science and machine learning by assessing the use of convolutional Deep Learning climate MOS approaches and designing our novel climate MOS architecture ConvMOS which considers the nature of typical errors present in precipitation estimates of climate models: (i) location-specific errors stemming from poor grid point representation of land surface characteristics, e.g. topography (Paeth 2011) or great lakes (Samuelsson et al. 2010) and (ii) systematic errors originating from the use of simplified climate processes, as is often the case for cloud and rainfall formation (Paeth 2011). To efficiently reduce both types of errors, ConvMOS—as shown in Fig. 1—combines per-location model parameters, which learn to reduce errors specific to a location, and global model parameters, which learn spatial precipitation patterns to effectively reduce systematic errors in climate model outputs. Our architecture composition studies (Sect. 5.2 and Appendix A.2) show that such parameter combinations improve climate MOS performance in practice. We also consider and evaluate other popular CNN architectures for climate MOS, namely ResNets (He et al. 2016) and U-Net (Ronneberger et al. 2015).

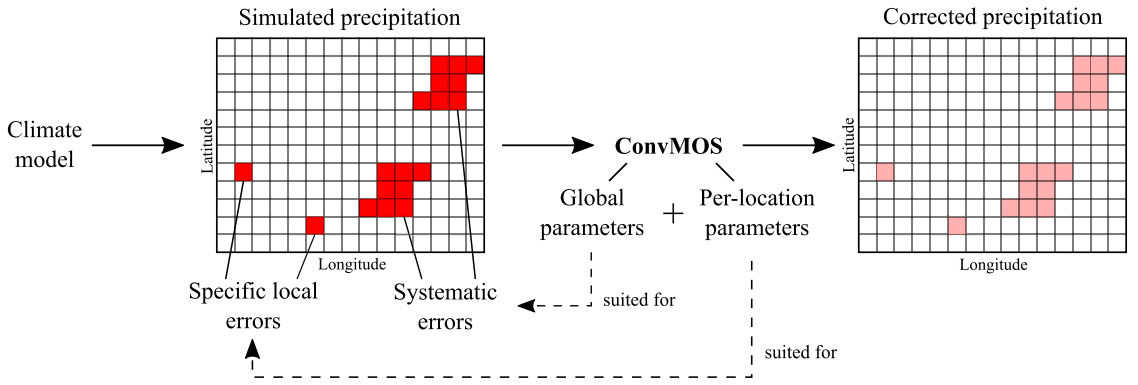


Fig. 1 ConvMOS: Systematic and location-specific errors in climate model outputs are reduced with our Deep Learning architecture that combines global and per-location parameters

We apply the approaches to correcting simulated precipitation of the RCM REMO (Majewski 1991; Jacob 2001; Jacob et al. 2001) and show that ConvMOS models reduce errors considerably, providing significantly better performance than the commonly used MOS approaches local Linear Regression (Eden and Widmann 2014), non-local Principal Component Regression (Eden and Widmann 2014), and non-local Random Forest (Sa'adi et al. 2017; Noor et al. 2019) in most cases. Additionally, we find that ConvMOS models typically perform better in comparison to plain ResNets or U-Net. Our results also show that all considered non-linear Deep Learning models underestimate the number of extreme precipitation events more than REMO and linear approaches. To alleviate this, we train ConvMOS models specialized towards estimating extreme precipitation events with the imbalanced regression method DenseLoss (Steininger et al. 2021), showing that such MOS models are better at estimating extreme precipitation events. Additional analysis is provided in the Appendix, where we analyze the training duration of the considered MOS techniques as well as MOS results over time. For this, we find no clear temporal error trends in our setting, suggesting that MOS approaches do not necessarily have to be updated over time. While we validated our approach on climate MOS, we argue that aspects of the ConvMOS architecture may also be beneficial for other applications with geospatial data, which is especially common in environmental tasks. Code and REMO data is available.[1]

We make the following contributions:

- We present a novel convolutional Deep Learning architecture for climate MOS *ConvMOS*, consisting of *local* and *global* network modules.
- We show with architecture composition studies (Sect. 5.2 and Appendix A.2) that the combination of per-location and global model parameters does indeed improve climate MOS performance.
- We compare ConvMOS to three commonly used climate MOS approaches and two popular CNN models, finding that our approach performs significantly better in most metrics.

[1] https://github.com/SteiMi/convmos An early version of this work was presented at NeurIPS 2020 Tackling Climate Change with Machine Learning Workshop (Steininger et al. 2020).

- We assess ConvMOS models specialized at estimating extreme precipitation events with the imbalanced regression method DenseLoss to allow for improved estimates for extreme events.

In Sect. 2 we discuss related research. Sect. 3 describes the data we used. We describe our proposed ConvMOS architecture for climate MOS in Sect. 4. In Sect. 5 we describe our experimental evaluation and its results. Sects. 6, and 7 discuss this work and consider its broader impact, respectively. Finally, Sect. 8 provides a conclusion.

2 Related work

The following introduces related prior work on spatio-temporal modeling, the climate MOS task considered in this work, and fully convolutional models which are related to the architecture proposed in this work.

2.1 Spatio-temporal modeling

In this work, we consider a combination of a climate model with machine learning techniques in order to provide spatio-temporal predictions of precipitation. While this is a standard approach in this particular domain, there are also other approaches to spatio-temporal modeling.

One approach is modeling spatio-temporal autocorrelation. Specific techniques include LASSO-VAR (Cavalcante et al. 2017), training Multilayer Perceptrons (MLPs) with entropy-based criteria (Ceci et al. 2019), or suitable feature extraction techniques in conjunction with tree models (Corizzo et al. 2021). There are also models which combine non-parametric tree models with parametric models for distribution tails in order to improve forecasting of extreme values (Gonçalves et al. 2021), which is similar in goal but different in technique to our experiment using a sample weighting technique for better extreme value estimation in Sect. 5.7.

Spatio-temporal forecasts are also often modeled with Deep Learning in domains like air pollution prediction, with approaches that combine temporal LSTM (Long Short-Term Memory) (Hochreiter and Schmidhuber 1997) layers with, for example, spatial attention (Shi et al. 2021), nearest neighbor approaches (Qin et al. 2019), or convolutional neural networks (CNNs) (Zhang et al. 2020). This combination of different model types for spatial and temporal aspects bears some resemblance to the approach proposed in this work, where local and global model parameters are combined to model different spatial aspects (location-specific and global, systematic errors).

The difference between the climate MOS task considered in this work and the aforementioned spatio-temporal modeling approaches is, that, strictly speaking, we do not consider climate MOS to be a forecasting task from a machine-learning-view. The temporal dynamics required for forecasts are entirely handled by the climate model. A MOS approach does not directly need to forecast future states, but only adjust the current state provided by the climate model. One may incorporate the time dimension in climate MOS approaches, but it is uncommon and may not necessarily be

beneficial, which is why this work focuses on traditional non-temporal climate MOS. Nonetheless, spatio-temporal models may benefit from also consider a combination of global and local parameters for their spatial and maybe even temporal parts in order to efficiently learn both global, systematic and location- or time-specific patterns.

2.2 Climate model output statistics

There are two approaches to climate MOS—distribution-wise and event-wise MOS. Distribution-wise MOS corrects the simulated variable's distribution by mapping distribution characteristics (e.g. means) to the observed distribution. Event-wise MOS links simulated and observed time series through statistical models, which generally performs better than distribution-wise MOS (Eden and Widmann 2014). Thus, this work considers event-wise MOS.

A simple approach used by Eden and Widmann (2014) is local Linear Regression where an individual Linear Regression is fitted per location of interest, which has shown to work reasonably well. Most other works propose non-local MOS approaches, where for each location the MOS is aware of climatic conditions at nearby locations. This can lead to a large number of predictors for the MOS, which is why dimensionality reduction techniques, e.g. principal component analysis (PCA), are often applied (Paeth 2011; Eden and Widmann 2014; Sa'adi et al. 2017; Noor et al. 2019). Non-local MOS has been done with a range of machine learning models namely Linear or Principal Component Regression (Paeth 2011; Eden and Widmann 2014), Random Forests (RFs) (Sa'adi et al. 2017; Noor et al. 2019), Support Vector Machines (SVMs) (Sa'adi et al. 2017; Pour et al. 2018; Ahmed et al. 2019), and Multilayer Perceptrons (Moghim and Bras 2017).

While these methods have proven to be effective, we believe that there is considerable potential in exploring advanced Deep Learning techniques. Especially CNNs (LeCun et al. 1998) have shown proficiency in tasks with geospatial data, where each input "pixel" relates to a geographic location on Earth and provides information on the state there like prior precipitation for precipitation forecasts (Shi et al. 2017) or land-usage for air pollution estimation (Steininger et al. 2020). This indicates potential for novel non-local climate MOS with this type of neural network.

2.3 Fully convolutional networks

A core aspect of the ConvMOS architecture is the use of fully convolutional networks. These are neural networks that consist solely of convolutional layers.

Fully convolutional networks were first introduced for semantic segmentation of images in the computer vision domain (Long et al. 2015). They are useful for tasks where both the input and the output are image-like, meaning that pixels or cells are arranged in a grid. This is the case in computer vision tasks like semantic segmentation or instance segmentation (He et al. 2017). A particularly notable fully convolutional network is U-Net (Ronneberger et al. 2015) that was proposed for biomedical image segmentation and has been applied to many problems like image-to-image translation since (Kandel et al. 2020).

Fully convolutional networks are also suitable for geospatial environmental machine learning tasks like climate MOS, since the locations of a study area can be arranged in an image-like grid with the different environmental variables (e.g. precipitation) being channels of this image. One domain where they have shown good results is statistical downscaling of climate data by improving its spatial resolution through fully convolutional super-resolution CNNs (Vandal et al. 2017; Liu et al. 2020). Similarly, fully convolutional networks have been used successfully for precipitation nowcasting, which is short-term forecasting of rainfall (Agrawal et al. 2019). These positive results for similarly structured data suggests that this model type can also be beneficial for climate MOS. We believe that their ability to learn spatial patterns is also well suited for efficiently reducing systematic errors in climate models. Recent work outside of the climate domain in the related field of post-processing ensemble weather forecasts has also shown promising results by applying fully convolutional CNNs and locally connected networks that are not translation invariant (Grönquist et al. 2021). Thus, using CNNs in combination with per-location model parameters, which can reduce location-specific errors, is a promising approach for use in climate MOS.

3 Dataset

For evaluation we use the model and observational data presented next.

Model Data We use daily data of the regional climate model (RCM) REMO (hydrostatic version REMO2015) (Majewski 1991; Jacob 2001; Jacob et al. 2001) for the period 2000 to 2015. REMO is based on the Europa Modell (Majewski 1991) and the model physics of the GCM ECHAM4 (Roeckner 1996) with further improvements (e.g. Hagemann (2002); Semmler (2002); Kotlarski (2007)). The reanalysis ERA-Interim (0.75° × 0.75°) (Dee 2011; Berrisford et al. 2011) is used as forcing data, providing the lateral boundary conditions. The atmosphere's vertical resolution is represented by 27 hybrid levels with increasing distance to the atmosphere's top. In lower levels they follow the topography (Teichmann 2010). Our study area spans over an extended German region with 0.11° resolution covering the area from -1.43° to 22.22° E and 42.77° to 57.06° N (GER-11). This grid does not have 215 × 130 cells as one would think based on area and resolution but instead 121 × 121 cells since the grid is not axially parallel to latitudes or longitudes due to REMO's usage of rotated coordinates for numerical reasons (Lüthi and Heinzeller 2017). We use 23 MOS predictors (see Table 1), which all stem from REMO except for the elevation from the GTOPO dataset (0.009° × 0.009°) (DAAC 1996; Gesch et al. 1999). REMO also uses GTOPO's elevation.

Observational Data For observational data we use the gridded dataset E-OBS (Haylock et al. 2008) version 19.0e. It is based on an ensemble of interpolated station data and is therefore subject to some uncertainty, as station density varies in space and time (Cornes et al. 2018). Our predictand is E-OBS's daily precipitation sums at 0.1° resolution. Both model and observational data are interpolated bilinearly to the same 0.11° grid (Schulzweida 2019).

Table 1 MOS predictors

Predictor	Height levels	Predictor	Height levels
Temperature (K)	2 m a. s.; 100;200;500;850;950 hPa	Geopotential height (m)	100; 200; 500; 850; 950 hPa
Min. temperature (K)	2 m a. s.	Total precipitation (mm)	–
Max. temperature (K)	2 m a. s.	Specific humidity (kg kg^{-1})	100; 200; 500; 850; 950 hPa
U-wind ($m\,s^{-1}$)	10 m a. s.	Sea level pressure (Pa)	–
V-wind ($m\,s^{-1}$)	10 m a. s.	Elevation (m)	–

Total precipitation is the sum of snowfall, convective, and large scale precipitation. "a. s." stands for "above surface"

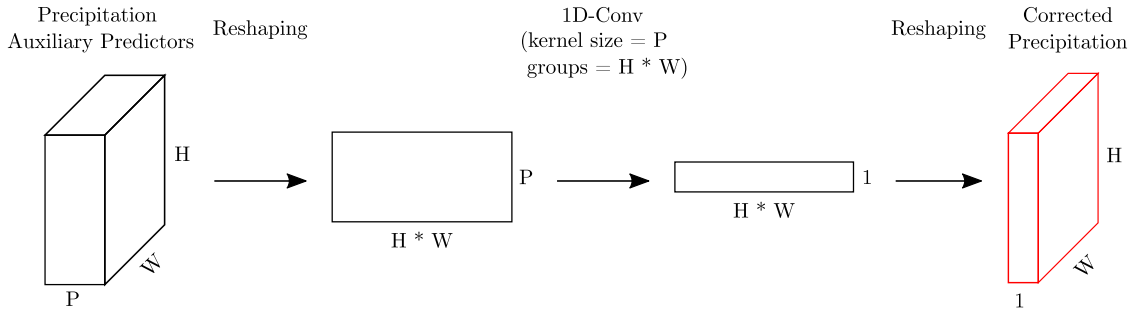


Fig. 2 Local network module's structure. H and W represent study area height and width. P is the number of predictors. Depiction is not to scale

4 ConvMOS

To explore the combination of global and per-location model parameters with CNNs as MOS we propose the architecture ConvMOS.

4.1 Idea

The basic idea of ConvMOS is derived from two sources of error in climate models: First, location-specific errors which often stem from poor grid point representation of topography. This representation can lead to abrupt topographic elevation, potentially influencing processes affecting precipitation (Paeth 2011; Eden and Widmann 2014). Second, systematic errors originating from parameterization, which replaces too complex or too small-scale processes with simpler variants. Cloud and rainfall formation is based on parameterization, leading to precipitation overestimation over land (Paeth 2011).

To efficiently reduce both types of errors, we propose a model consisting of both per-location model parameters, which can learn the characteristics of a specific location, and global model parameters, which can learn spatial precipitation patterns to efficiently help reduce systematic errors in climate models. Thus, we define two modules: local network and global network.

4.2 Local network

The local network module contains individual model parameters for each location in the study area, allowing it to reduce specific local errors. For ease of integration into the neural network architecture, we do not use a separate model (e.g. a linear regression) per location. Instead, we implement this through reshaping and a linearly activated 1D CNN, as is depicted in Fig. 2. The input at each time of size (predictors, height, width) is first reshaped so it has the dimensions (height $*$ width, predictors). In conjunction with setting the kernel size equal to the number of predictors, this allows us to group the convolution for each input channel (i.e. each location) so that each location is convolved with its own set of filters for all predictors. Thus, each location has its own model parameters, in which location characteristics can be encoded. The 1D CNN output is of shape (height $*$ width, 1) which we reshape to (1, height, width), giving

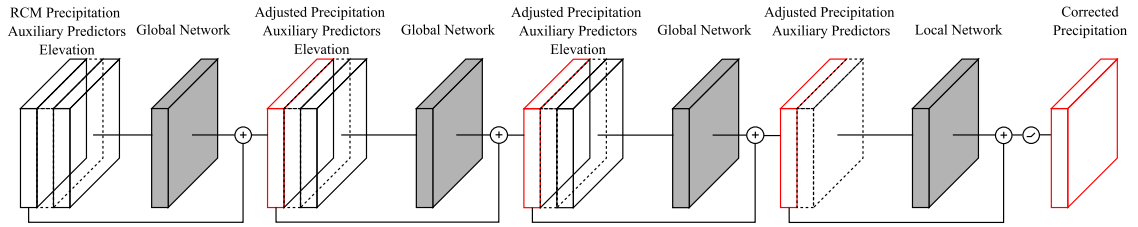


Fig. 3 ConvMOS architecture with the composition ConvMOS-gggl, having three global and one local network module

us the output of the local network module. This output can be interpreted as a grid with per-location precipitation residuals. This module is not provided with elevation data as it would be static across all times for each location.

This approach allows us to integrate per-location model parameters seamlessly into a Deep Learning model. The naive alternative of using separate models per location is harder to implement concurrently within an Deep Learning architecture running on a GPU. For Deep Learning libraries such as PyTorch (Paszke et al. 2019), which we use in this work, our module is simply another convolutional layer. This allows for efficient training and inference.

4.3 Global network

The global network learns spatial patterns in precipitation and other predictors. This can be done efficiently with CNNs (Vandal et al. 2017). The module contains a 2D CNN with four layers which should be well suited for learning useful filters which can reduce systematic errors across the study area. In addition to the local modules' predictors, the global network is also provided with elevation data for each location. In contrast to the per-location model parameters, this information is not static for the filters of the 2D CNN since the filters are applied for all locations across the study area. Starting from the first layer, the layers have 4, 8, 16, and 1 filters and kernel sizes of 9, 1, 5, and 3, respectively. Each convolutional layer has its padding parameter set to half its kernel size (rounded down to the nearest whole number) which leads to each layer's output having the same width and height as its input. All layers use a stride and a dilation of 1. The first three layers use the ReLU (Nair and Hinton 2010) activation function while the last layer is activated linearly. As with the local network, this module also outputs a grid of precipitation residuals.

4.4 Architecture

The ConvMOS architecture consists of sequentially concatenated instances of global and local network modules. Figure 3 depicts an example of a ConvMOS model. ConvMOS expects a 3D input with dimensions (predictors, height, width) for each time step. The data is sequentially passed through the modules (depicted in gray) where each module adjusts the precipitation input with the goal of reducing the error. The architecture employs so called "shortcut connections" for each module where each module's output is added to its precipitation input, which eases training for neural net-

works (He et al. 2016). In this work, we employ the depicted model with three global networks followed by a local network module, which is the result of our architecture composition study described in Sect. 5.2. We call this exact architecture composition ConvMOS-gggl. The global networks aim to reduce any systematic errors across the study area. Finally, the local network corrects any specific local errors and makes sure that the systematic corrections of the global network are not introducing new local errors. As precipitation cannot be negative we use a ReLU after the final shortcut connection to force positive values. The architecture is fitted with the Adam optimizer (Kingma and Ba 2014), the mean squared error (MSE) as the loss function, a learning rate of 0.001, and a batch size of 128. Only errors at locations where observational data is available were incorporated for the MSE. Training is conducted for at most 100000 epochs. Early stopping is used to stop training when the validation MSE is not improving for more than 40 consecutive epochs, preventing considerable overfitting (Caruana et al. 2001).

5 Experiment

To evaluate ConvMOS models, we apply them to the data described in Sect. 3. After defining our experimental setup, we evaluate our hypothesis regarding the benefit of combined per-location model parameters and global model parameters while also finding ConvMOS's best architecture composition for use in the experiment. We also apply standard ResNet and U-Net CNN models in addition to three commonly used MOS approaches, a local Linear Regression, a non-local Principal Component Regression approach and a non-local Random Forest method, for comparison and evaluate them for general and seasonal performance. Thereafter, we assess ConvMOS models specialized towards estimating extreme precipitation events using the imbalanced regression method DenseLoss. Additional analysis can be found in the Appendix, where we analyze the training duration of the considered MOS approaches and evaluate MOS results over time, finding no clear temporal error trends which suggests that—at least for the climate model and timespan considered in this work—MOS approaches do not necessarily have to be updated over time.

5.1 Experimental setup

We split the 16 years of daily data into a training (2000–2009), a validation (2010), and a test set (2011–2015). All predictors are standardized based on the training set so that they have a mean of zero and a standard deviation of one. Target values are not standardized and metrics are thus also computed on non-standardized data. The hyperparameter values presented in this work for the local and global network modules of our ConvMOS architecture were selected based on preliminary tests using the validation set. For evaluation, we use a number of common MOS metrics, namely root-mean-squared error (RMSE), normalized RMSE (NRMSE), Pearson Correlation, Skill score (Perkins et al. 2007), R^2, and Bias to assess different performance aspects. NRMSE divides the RMSE for each location in the study area by the dif-

ference between the maximum and minimum observed precipitation there, which we then multiply by 100 to receive a percentage. Skill score measures the common area between the probability density function of the observed precipitation and the simulated precipitation. To this end, data is binned (we use bins of 1 mm width as Perkins did) and the Skill score is both distributions' cumulative minimum value of each binned value. Thus, a perfect Skill score would be 1. R^2 describes the proportion of variance explained by a model with 1 being a perfect score. Models with R^2 lower than 0 fit worse than the data's mean. The Bias metric is the mean error. A positive value indicates overestimation of precipitation while a negative value indicates the opposite. MOS approaches with non-deterministic fitting methods, i.e. ConvMOS, ResNets, U-Net and the non-local Random Forest, are trained 20 times since performance may differ per fitted instance. All reported mean Correlations use Fisher's z-transformation (Silver and Dunlap 1987). When we report significant differences in the following, we confirmed this with a Wilcoxon signed-rank test (Wilcoxon 1945) and a significance level of 0.05.

5.2 Architecture composition study

The key idea behind ConvMOS is the combination of per-location model parameters and global model parameters which is why the architecture allows for different combinations of sequentially connected local and global network modules. In order to test whether this combination is beneficial and to find the best module arrangement we evaluate a number of composition candidates. We train 20 instances per composition on the training set and test them on the validation set. To allow for early stopping we remove the 2009 data from the training set, evaluate the model after each epoch on this data and stop training when the MSE in 2009 does not improve for more than 40 epochs in a row.

Table 2 shows mean metrics on the validation set for all study area locations available in observational data (i.e. land points) of each architecture composition sorted by RMSE. The architecture ConvMOS-gggl shows the best performance, surpassing all other tested compositions in terms of RMSE, NRMSE, Correlation, and R^2. Compared to ConvMOS-glgl with the second lowest RMSE, ConvMOS-gggl's RMSE and Bias are not significantly different but its NRMSE, Correlation and R^2 are significantly better. ConvMOS-gggl's Skill score is not significantly different from the best model for that metric (ConvMOS-glll) as well as its Bias, which is also not significantly different from the model with lowest Bias (ConvMOS-ggl). Overall, we consider ConvMOS-gggl to provide the best performance, which is why we choose this composition for our experiment. We find that, considering the results of ConvMOS-ggl and ConvMOS-gl, additional global network modules at the model's front reduces errors further, presumably since more complex spatial patterns can be learned. ConvMOS-ggl performs significantly better than ConvMOS-gl in all metrics. ConvMOS-gggl is significantly better than ConvMOS-ggl only in RMSE, NRMSE, and Correlation. This suggests diminishing improvements with more global modules. The results also show that the key idea behind ConvMOS—the combination of per-location and global model parameters—can indeed improve performance in terms of RMSE, NRMSE,

Table 2 Mean validation set metrics per architecture composition sorted by RMSE (left side < right side), rounded to three decimal places

Mod.	RMSE	NRMSE	Cor.	Skill	R^2	Bias	Mod.	RMSE	NRMSE	Cor.	Skill	R^2	Bias
gggl	**3.282**	**8.817**	**.748**	.843	**0.511**	.035	lgll	3.349	9.008	.735	.824	0.489	.053
glgl	3.292	8.863	.746	.838	0.506	.067	gl	3.351	8.976	.735	.837	0.496	.051
gglg	3.298	9.442	.746	.825	−0.389	.064	llgl	3.351	9.032	.734	.817	0.483	.054
ggl	3.299	8.850	.745	.844	0.509	**.015**	llgg	3.356	9.652	.737	.821	−0.340	.078
ggll	3.302	8.861	.745	.844	0.507	.034	gggg	3.362	11.327	.744	.827	−7.511	.047
glgg	3.306	9.746	.747	.825	−1.058	.067	ggg	3.365	11.345	.744	.826	−7.555	.054
lggl	3.309	8.980	.743	.827	0.462	.054	llg	3.381	9.488	.730	.813	0.047	.091
glg	3.310	9.471	.745	.825	−0.419	.061	lg	3.382	9.579	.731	.816	−0.150	.077
gllg	3.318	9.445	.743	.825	−0.312	.071	lllg	3.386	9.467	.729	.818	0.094	.064
lglg	3.334	9.512	.740	.829	−0.235	.066	gg	3.404	11.543	.737	.824	−8.338	.085
lggg	3.344	9.885	.740	.824	−1.012	.084	g	3.486	11.701	.722	.816	−8.327	.101
lgg	3.347	9.717	.739	.822	−0.574	.066	llll	3.786	10.044	.649	.795	0.375	.189
lgl	3.347	9.004	.735	.821	0.489	.061	lll	3.790	10.054	.648	.793	0.374	.195
glll	3.348	8.968	.736	**.846**	0.496	.046	ll	3.792	10.060	.647	.792	0.373	.197
gll	3.348	8.969	.736	.843	0.496	.036	l	3.802	10.085	.644	.784	0.371	.202

Bold values indicate the best value

"Mod." is modules, "Cor." is Correlation, "g"/"l" is global/local network. RMSE and Bias are in mm, NRMSE is in %

Correlation, and R^2. The best architecture composition ConvMOS-gggl consists of a combination of different modules. Furthermore, it performs significantly better in terms of the aforementioned four metrics than the best composition consisting of solely local or global modules, namely ConvMOS-gggg. Compositions consisting solely of local modules or global modules typically perform worse than combinations of both. Additionally, we find that having a local network module as the final module provides relatively good NRMSE and R^2 values. We hypothesize that the global module's filters adjust precipitation in a similar way everywhere, leading to low performance for these metrics in some areas, e.g. when there is only relatively minor precipitation. An additional architecture composition study with U-Nets as global modules in the Appendix (see Appendix A.2) further confirms most findings presented here.

5.3 Standard climate MOS approaches

For comparison, we also evaluate standard climate MOS approaches. Similar to prior work (Paeth 2011; Sa'adi et al. 2017; Noor et al. 2019), we preprocess the standard MOS methods' predictors to reduce dimensionality and remove potentially unhelpful information. Like Sa'adi et al. (2017) and Noor et al. (2019) we use supervised PCA (Bair et al. 2006). For each location, we select the best predictors based on a univariate regression. Local MOS models choose from 23 predictors for a specific location while non-local models have another 23 predictors per considered nearby location (i.e. $11 \times 11 \times 23 = 2783$ predictors when considering locations at most 5 cells away). The number of retained predictors is set according to an exhaustive grid search at each location that considers choosing the 1 to 30 best predictors with our validation data. Then, PCA reduces the dimensionality of these predictors, keeping the first components that explain at least 95 % of the variance (Sa'adi et al. 2017). All non-Deep-Learning MOS methods described in the following use this preprocessing scheme.

Local Linear Regression (Lin) For each cell in the study area, a separate Linear Regression is fitted where simulated precipitation is the predictor and observed precipitation is the predictand. This approach is local in that each Linear Regression is unaware of conditions in nearby cells (Eden and Widmann 2014).

Non-local Principal Component Regression (NL PCR) Instead of only using large-scale conditions at a specific location for a Linear Regression, we provide all available predictors at each nearby location (at most 5 cells away in either direction) on the grid. This is feasible with the help of the supervised PCA which reduces the dimensionality of the predictors (Eden and Widmann 2014).

Non-local Random Forest (NL RF) For the non-local Random Forest, we provide all available predictors of each location ± 5 cells away, as with NL PCR. The supervised PCA applied for preprocessing is also what Sa'adi et al. (2017) and Noor et al. (2019) used. Each location in our study area has its own RF instance for MOS which uses scikit-learn's RF (Pedregosa et al. 2011). Since RF performance depends considerably on its hyperparameters, we look for optimal values with a random search. For each cell we train 20 RF instances on the training set with hyperparameter values sampled randomly from the search space shown in Table 6. Each instance is evaluated on the validation set. The RF instance with the best R^2 is then applied on the test set.

5.4 Standard deep learning approaches

To further put our results in perspective, we also apply some common Deep Learning architectures. Suitable architectures allow mapping an input image to a new output image of the same size since this is structurally similar to our task of mapping an input climate to a precipitation output with the same spatial dimensions. In our experiment, we consider two commonly used architectures, namely ResNet (He et al. 2016) and U-Net (Ronneberger et al. 2015).

ResNet ResNets are popular models in computer vision which is why it is interesting to see how such a general architecture fares for climate MOS. ResNets are available with different numbers of layers. In our experiment, we used ResNet18, ResNet34, ResNet50, and ResNet101. We omit ResNet152 as its memory requirements are too large for most GPUs available to us when trained on our task and we also found no performance gains between larger and smaller ResNets anyways. The ResNets are adapted for our task by changing the number of input features in the first convolutional layer from 3 to 23 (one per predictor), removing the softmax activation necessary for classification, and replacing the final fully connected layer with one that maps to 121 x 121 (height x width) outputs. Training is conducted in the same way as for ConvMOS (i.e. same learning rate, optimizer, early stopping, loss, batch size).

U-Net Another important architecture for image-to-image tasks is U-Net. This architecture has already shown its proficiency in the related task of post-processing ensemble weather forecasts (Grönquist et al. 2021). Because of this similarity, we use their U-Net variant that differs from the standard U-Net in a few aspects: (i) Up-convolutions are replaced with bilinear interpolation followed by a 3 x 3 convolution with stride 1 to avoid checkerboard artefacts. (ii) U-Net's five levels are reduced to three levels to avoid overfitting. (iii) The number of filters per convolution are halved as they observed no improved performance with more filters.

Training is conducted in the same way as for ConvMOS (i.e. same learning rate, optimizer, early stopping, loss, batch size).

We also evaluate the use of this U-Net within the ConvMOS architecture by using it as a global network module instead of the one presented in Sect. 4.3. For this approach, we sequentially connect one global network module (here a U-Net) and one local network module, which is the resulting composition of the architecture composition study in the Appendix (see Appendix A.2). This is similar to the model proposed by Grönquist et al. (2021) for their weather forecasting task but with a ConvMOS local network module after the U-Net instead of their locally connected network. This approach is denoted as *ConvMOS-UNet* or short *CM-UNet* in the following.

5.5 Results

Table 3 shows mean metrics on the test set for all study area locations available in observational data. All MOS approaches improve all metrics considerably when compared to applying no MOS, except for the Skill score. This suggests that REMO's precipitation distribution at land locations is already rather close to that of the observations with a Skill score of 0.93 and can barely be improved by MOS methods. All

Table 3 Test set mean metrics for all locations having observational data

Metric MOS	RMSE (mm)	NRMSE (%)	Cor.	Skill	R^2	Bias (mm)
REMO raw	5.32	15.83	0.49	**0.91**	−28.24	0.31
Lin	3.51	8.03	0.58	0.47	0.33	−0.03
NL PCR	3.37	7.80	0.62	0.81	0.36	0.02
NL RF	3.39 ± 0.00	7.82 ± 0.00	0.61	0.82	0.36 ± 0.00	0.03 ± 0.00
ResNet18	3.03 ± 0.01	7.04 ± 0.03	0.71	0.60	0.47 ± 0.01	−0.06 ± 0.07
ResNet34	3.06 ± 0.02	7.10 ± 0.04	0.71	0.61	0.46 ± 0.01	−0.07 ± 0.09
ResNet50	3.04 ± 0.01	7.05 ± 0.03	0.71	0.61	0.47 ± 0.00	−0.10 ± 0.10
ResNet101	3.03 ± 0.02	7.04 ± 0.04	0.71	0.64	0.47 ± 0.01	−0.04 ± 0.08
U-Net	2.97 ± 0.02	8.37 ± 0.12	**0.74**	0.82	−5.60 ± 0.88	−0.03 ± 0.08
CM-UNet	**2.92 ± 0.01**	7.01 ± 0.11	**0.74**	0.70	0.13 ± 0.22	**0.01 ± 0.10**
ConvMOS	2.93 ± 0.02	**6.77 ± 0.05**	0.73	0.89	**0.51 ± 0.02**	−0.10 ± 0.05

Bold values indicate the best value
Values rounded to two decimal places. Std. dev. for Correlation (always 0.00) and Skill score (between 0.00 and 0.03) omitted for brevity

Deep-Learning-based MOS approaches perform better than standard approaches in terms of RMSE, NRMSE, Correlation and R^2, except for U-Net's NRMSE as well as U-Net's and CM-UNet's R^2. We find that U-Net and, to a lesser extent, CM-UNet struggle at some locations as can be seen in the Appendix' Fig. 7 for NRMSE. These low performance locations typically have very low precipitation, with which these models in particular have issues. The two ConvMOS models combining local and global model weights—CM-UNet and ConvMOS—tend to perform best. CM-UNet provides significantly better RMSE than all other approaches except for ConvMOS. CM-UNet's correlation is also significantly better than all other MOS methods except for U-Net, while ConvMOS is also only closely behind. For NRMSE, Skill score, and R^2, ConvMOS is significantly better than all other MOS approaches. This indicates that ConvMOS-based approaches can estimate precipitation more accurately than all considered comparison methods. ConvMOS's Skill score is close but still reduced slightly by 0.02 compared to REMO's. ConvMOS shows less Bias than REMO but it seems to have a tendency to underestimate precipitation as most approaches do. CM-UNet tends to show the lowest Bias. We also ran this experiment with precipitation as the only climate predictor as some prior work has done (Eden and Widmann 2014; Noor et al. 2019) but found all considered methods to perform worse without additional predictors.

Figure 4 visualizes RMSEs for all locations with observational data across the study area for all assessed approaches using each method's best instance with regard to test RMSE. Similarly as in Table 3, all MOS methods can reduce errors from the original REMO output. Especially precipitation in the Alps and other mountainous regions is improved considerably. We find that CNN approaches tend to provide lower errors compared to other MOS methods in general but also for seemingly difficult areas. All standard MOS approaches show a bright yellow spot near the border between Italy and

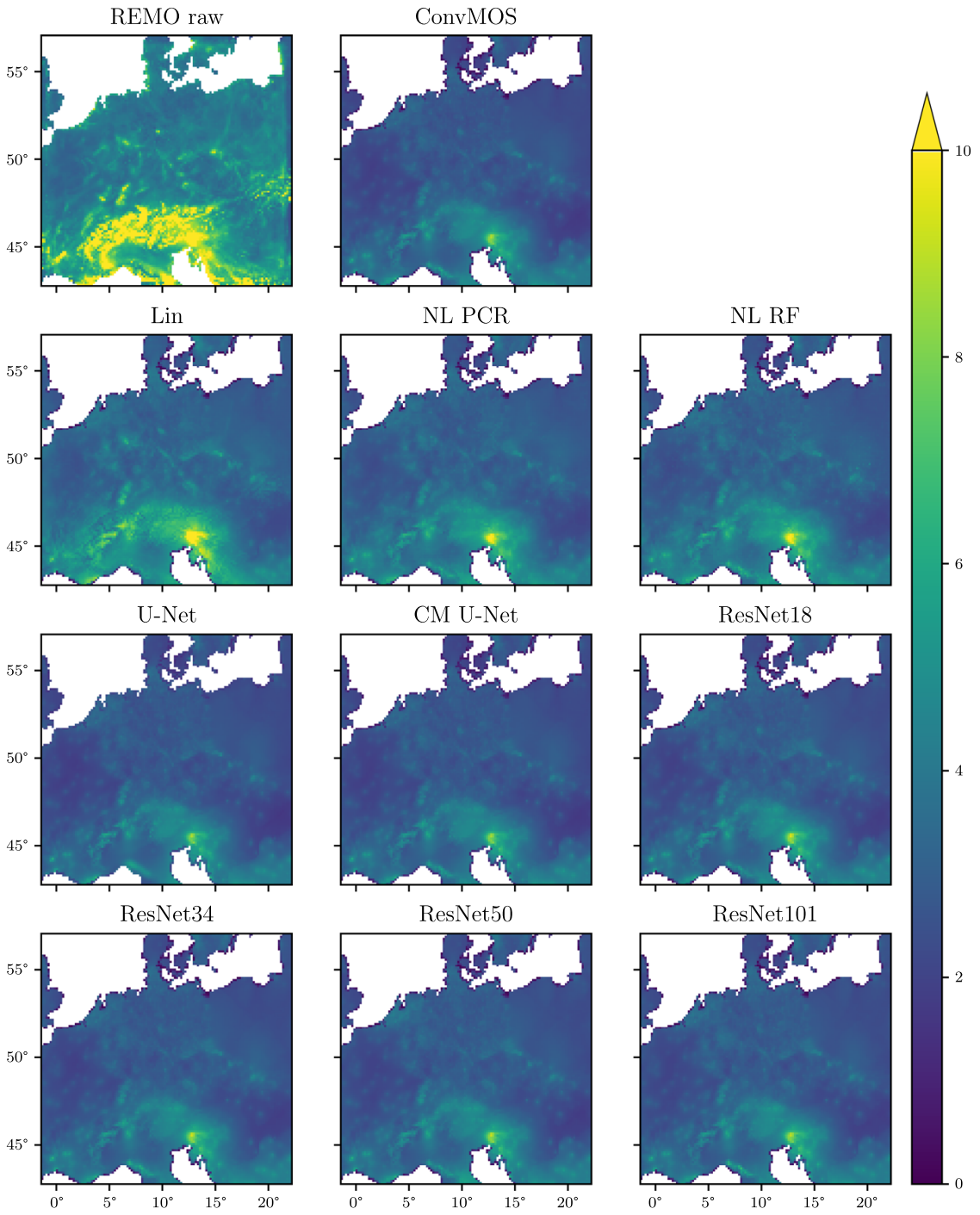


Fig. 4 RMSE of precipitation in mm for the test set across the study area. *Note* REMO raw has cells with far larger RMSE than 10 mm but we limited the colorbar's extent for better visibility of general performance

Slovenia indicating high error and difficulty there, that is less pronounced for CNN models. In the observational E-OBS data for this area we noticed that there tends to be higher precipitation during the test time frame compared to the training time frame. We hypothesize that especially the standard MOS approaches have difficulties due to this shift in the precipitation distribution there.

Figure 5 depicts the daily precipitation distributions on the test set for all locations with observational data for E-OBS's observed precipitation, REMO's precipitation,

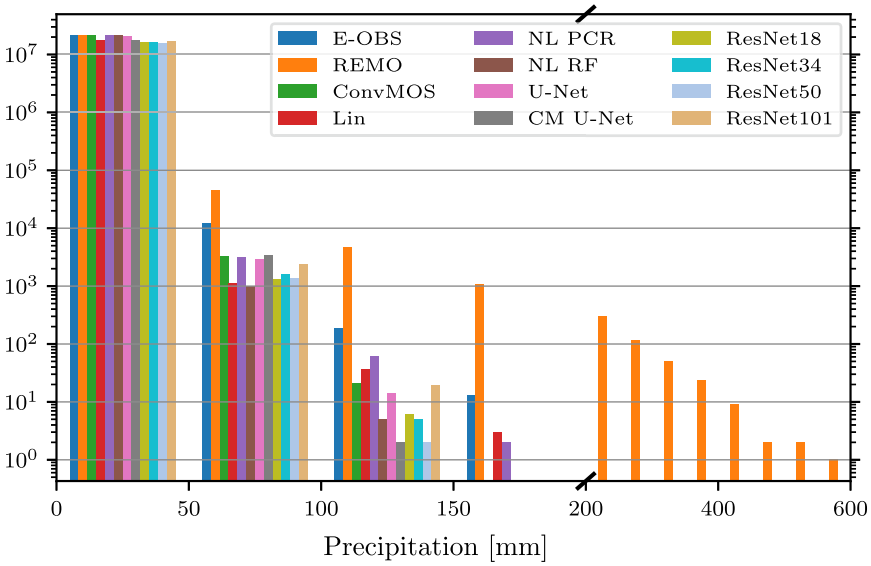


Fig. 5 Daily precipitation distributions on the test set. The 12 bins are 50 mm wide, starting at 0 mm. The y-axis (number of events) is scaled logarithmically and the x-axis (precipitation) is compressed over 200 mm for brevity

and the outputs of each MOS method's best model instance (i.e. lowest test RMSE). It shows that REMO often simulates considerably more precipitation than ever observed despite the good mean Skill score per location, showing the weaknesses of its hydrological cycle. All MOS approaches underestimate the number of high precipitation events ($\geq$ 50 mm). NL RF and all Deep Learning models are particularly conservative about their estimates, showing considerably fewer events with more than 100 mm than both linear MOS approaches and E-OBS. This indicates room for improvement when considering relatively rare extreme precipitation events with non-linear MOS.

5.6 Seasonal results

To assess whether the MOS approaches fitted with training data covering entire years exhibit seasonal anomalies, we also evaluate them per season. Table 4 shows the mean RMSE per season on the test set for all study area locations available in observational data. The seasons are DJF (December–February), MAM (March–May), JJA (June–August), and SON (September–November).

The seasonal results show that all MOS methods reduce errors across the year. REMO seems to have more problems estimating precipitation during summer and autumn (i.e. JJA and SON) for this study area which also results in larger RMSE of MOS outputs in these seasons. As with the overall RMSE, CM-UNet and ConvMOS are providing the best RMSE across the seasons with similarly low standard deviations. ConvMOS is slightly better in the more difficult JJA and SON seasons while CM-UNet is best during DJF and MAM. This difference is statistically significant for all seasons. All Deep Learning models that do not combine local and global weights are better than the standard approaches but worse than ConvMOS and CM-UNet. NL PCR and NL RF have similar performance and both tend to perform better than Lin.

Table 4 Seasonal RMSE in mm for all locations with observational data

Season MOS	DJF	MAM	JJA	SON
REMO raw	4.59	4.50	6.28	5.13
Lin	2.64	3.04	4.22	3.74
NL PCR	2.44	2.89	4.20	3.48
NL RF	2.48 ± 0.00	2.94 ± 0.00	4.18 ± 0.00	3.51 ± 0.00
ResNet18	2.14 ± 0.03	2.62 ± 0.01	3.77 ± 0.01	3.18 ± 0.02
ResNet34	2.18 ± 0.03	2.64 ± 0.01	3.79 ± 0.03	3.22 ± 0.02
ResNet50	2.13 ± 0.02	2.62 ± 0.01	3.77 ± 0.01	3.19 ± 0.02
ResNet101	2.13 ± 0.03	2.61 ± 0.02	3.77 ± 0.02	3.18 ± 0.02
U-Net	2.11 ± 0.02	2.54 ± 0.02	3.68 ± 0.03	3.09 ± 0.03
CM-UNet	$\mathbf{2.06 \pm 0.02}$	$\mathbf{2.51 \pm 0.01}$	3.64 ± 0.02	$\mathbf{3.04 \pm 0.02}$
ConvMOS	2.09 ± 0.02	2.52 ± 0.01	$\mathbf{3.63 \pm 0.02}$	$\mathbf{3.04 \pm 0.02}$

Bold values indicate the best value
Values rounded to two decimal places. "DJF" is December–February, "MAM" is March–May, "JJA" is June–August, "SON" is September–November

5.7 Focusing on extreme precipitation estimation

Our results show that non-linear models underestimate the number of extreme precipitation events more severely than REMO and linear approaches (see Fig. 5). These events can have negative effects on society and the environment like floods (Kundzewicz 2003), impact on plants (Zeppel et al. 2014) or increased disease spread (Chen et al. 2012). As such, it can be of interest to train models that perform particularly well for estimating the number and intensity of extreme events. Thus, we adapt ConvMOS-gggl to improve extreme precipitation estimation as it is among the best models in our experiment. In the following, we consider daily precipitation of at least 50 mm as extreme which is also the threshold at which the German Meteorological Service gives out a stage 3 precipitation warning for very dangerous weather (Deutscher Wetterdienst 2021).

One technique for training regression models with more emphasis on performance for rare data points in comparison to common data points is *DenseLoss* (Steininger et al. 2021). It estimates the target variable's density function from the training data points and gives each training data point a weight based on each sample's target value density. These weights are higher for samples in relatively rare parts of the target value range (i.e. extreme precipitation samples) in comparison to samples from more common parts of the target value range (i.e. precipitation closer to 0 mm). A sample's weight influences how much the error of that sample influences model training, leading to models better suited for estimating rare data points such as samples with extreme precipitation. The magnitude of weighting differences between samples with different rarity is configured through α. Through preliminary tests on the validation set, we found $\alpha = 1.0$ to provide the lowest RMSE for extreme samples which is why we set α to one. DenseLoss' minimal weight threshold ϵ is set to 10^{-6} as in the original paper. We modify the early stopping procedure to consider the validation MSE of extreme

samples only and train 20 instances of ConvMOS with DenseLoss, which we call ConvMOS-DL in the following. DenseLoss shifts the model's focus towards extreme precipitation events due to which we expect model performance for non-extreme samples to degrade to some extent while extreme data points are estimated more accurately.

To asses performance for extreme precipitation events, we split all samples into two bins, evaluating all occurrences of at least 50 mm separately from those with lower precipitation in the test set. The test set contains 12240 extreme and 21331874 non-extreme samples. Due to the rarity of extreme events we can not calculate meaningful mean metrics per cell but instead report mean metrics over all samples of a bin. In addition to the RMSE, we also evaluate how well a model can distinguish between extreme and non-extreme samples. To this end, we calculate a recall per bin and the balanced accuracy, which is defined as the mean of the extreme and non-extreme recalls. We consider a prediction accurate if it is lower than 50 mm for non-extreme samples and at least 50 mm for extreme samples. Table 5 shows RMSE and recall for REMO's raw output, ConvMOS, and ConvMOS-DL for non-extreme and extreme samples as well as the models' balanced accuracies. As expected, ConvMOS performs better in terms of both metrics for non-extreme data points in comparison to ConvMOS-DL and REMO raw, while the model using DenseLoss is still better than the raw REMO output. For extreme precipitation events, we see significantly better performance with ConvMOS-DL compared to ConvMOS. ConvMOS-DL can correctly identify on average 20.99 % of the extreme samples while ConvMOS only identifies 11.94 % correctly. REMO raw is closer to ConvMOS-DL's recall on extreme samples with 20.03 % but has considerably higher RMSE. When considering balanced accuracy, we find that ConvMOS-DL can distinguish best between extreme and non-extreme samples while REMO is similarly skilled in this aspect. Improved prediction of extreme values with DenseLoss can also be seen in a histogram, where the distribution is visibly closer to the observed precipitation (see Appendix). All in all, we find that DenseLoss can be used to train climate MOS models better suited for the analysis of extreme precipitation. Such models provide lower general performance but can distinguish better between extreme and non-extreme events while also showing lower errors for extreme precipitation events.

6 Discussion

In this work, we have shown that convolutional climate MOS and especially ConvMOS models can improve the quality of precipitation data significantly. However, we also found that especially non-linear approaches tended to perform poorly for the estimation of extreme precipitation events. We were able to alleviate this by training models specialized for extreme events with DenseLoss but ideally we could train models that perform well for both extreme and non-extreme precipitation events. Approaches to consider in the future for this may be uncertainty quantification methods which explicitly model uncertainty and, thus, may provide estimates that better follow the desired distribution (Abdar et al. 2021). It remains to be seen whether such techniques help the estimates' distribution to become closer to the real distribution while keeping metrics like RMSE at similarly low or even lower levels as reported here.

Table 5 Test set mean metrics for extreme/non-extreme observed precipitation ($\geq$ or $<$ 50 mm)

Metric	$<$ 50 mm		$\geq$ 50 mm		Balanced Acc. [%]
MOS	RMSE [mm]	Recall [%]	RMSE [mm]	Recall [%]	
REMO raw	5.75	99.77	47.48	20.03	59.90
ConvMOS	**2.94 ± 0.01**	**100.00 ± 0.00**	35.29 ± 0.76	11.94 ± 2.13	55.97 ± 1.07
ConvMOS-DL	4.15 ± 0.12	99.99 ± 0.00	**32.01 ± 0.64**	**20.99 ± 2.19**	**60.49 ± 1.09**

Bold values indicate the best value
"Acc." is accuracy. Values rounded to two decimal places

The MOS methods evaluated in this work only consider the spatial but not directly the temporal aspect of this task. The climate state at a particular time is dependent on the previous states and in our case only the climate model takes this into account. It is possible that including information of earlier time steps within the climate MOS models can help improve performance even further. It would therefore be interesting to consider this for future work.

As usual with machine learning techniques, it is often important to set suitable hyperparameters to achieve decent performance with a specific estimator. While it is feasibly possible to optimize the hyperparameters even for each location individually with the NL RF baseline, it is considerably more complex to tune Deep Learning models due to the enormous number of hyperparameters to consider and the dependencies between hyperparameters (e.g. CNN kernel sizes affect the output tensor shape, which can affect the structure of all following layers). For this reason, we only conducted limited hyperparameter tuning for ConvMOS and CM-UNet (e.g. architecture composition studies) and no tuning for the baseline ResNet and U-Net architectures. While the performance for all Deep Learning approaches and especially the ResNets and U-Nets may be further improved to some extent, this does not affect the main point of this work, namely that a combination of global and location-specific model parameters is beneficial as shown in both architecture composition studies. We furthermore believe that using pre-defined ResNets and U-Nets from prior work is an interesting baseline as these are likely models a practitioner would use, especially if the hardware and time is not available for more involved hyperparameter searches when conducting a climate study.

In contrast to reducing errors with climate MOS after running a climate model, a different approach to improving climate data is to directly reduce the source of errors in climate models. Uncertainties in climate models are primarily caused by the approximation of complex, high resolution processes through so-called parametrizations. To this end, there is work on learning better parametrizations with Deep Learning techniques, but they are not good enough yet to be used in practice (Rasp et al. 2018). Until these problems are solved, climate MOS methods like those considered in this work can be used as an effective tool for correcting climate model outputs.

7 Broader impact

The experiments conducted in this work consider climate MOS specifically and show that ConvMOS's combination of global and local model parameters are beneficial for the estimation quality. However, we believe that other domains may also benefit from aspects of ConvMOS's architecture. Location-specific parameters allow for the implicit encoding of a location's special characteristics during training, which we suspect to also be beneficial for other domains with geospatial data, where models like CNNs with their global model parameters are generally used on their own. Such data is common in environmental machine learning tasks like air pollution modeling or weather forecasting. For example, air pollution forecasting approaches like the one proposed by Zhang et al. (2020) use a CNN-based spatial feature extractor where each input "pixel" corresponds to a specific location that has its specific characteristics. We believe that the combination of the existing CNN-based model for the efficient

extraction of spatial features with a model containing per-location weights is likely to improve the overall model, as it is now able to encode location-specific characteristics that may be important for air pollution modeling.

Within the climate domain, this work provides a powerful new tool with ConvMOS. We hope to promote the application of ConvMOS through our publicly available code. This allows researchers conducting climate studies to apply our technique in order to provide them with more accurate data.

Besides the methodological and practical impact, we hope to foster more interest with our work in the data mining and machine learning community towards novel contributions for environmental tasks. Environmental issues like climate change are among the most pressing issues of our time and we believe that our community can provide important contributions for understanding, mitigation, and adaption of and to these processes, as is laid out in more detail in Rolnick et al. (2022).

8 Conclusion

In this work, we assessed convolutional Deep Learning climate MOS approaches and presented our ConvMOS architecture that is built specifically to reduce location-specific errors as well as systematic errors in climate model outputs. We applied ConvMOS models to the output of the RCM REMO in order to reduce errors in its simulated precipitation. In our architecture composition study, we showed that the combination of per-location model parameters and global model parameters is beneficial for MOS performance. Furthermore, our MOS approach is able to improve daily precipitation data considerably while also providing significantly better performance than three commonly used MOS approaches and plain ResNet and U-Net models in most cases. We also showed that issues of non-linear Deep Learning MOS for estimating extreme precipitation events can be alleviated by training models specialized for extreme events with the imbalanced regression method DenseLoss. Improvements in MOS allow for more accurate climate data especially at high spatial resolutions which allows us to better assess the effects of climate change. While ConvMOS is designed with climate MOS in mind, we believe that the architecture's combination of location-specific and global model parameters can also be beneficial for other tasks with geospatial data (e.g. air pollution modeling, weather forecasting), which opens interesting opportunities for future work.

Acknowledgements This research was conducted in the BigData@Geo project supported by the European Regional Development Fund (ERDF).

Funding Open Access funding enabled and organized by Projekt DEAL.

A Appendix

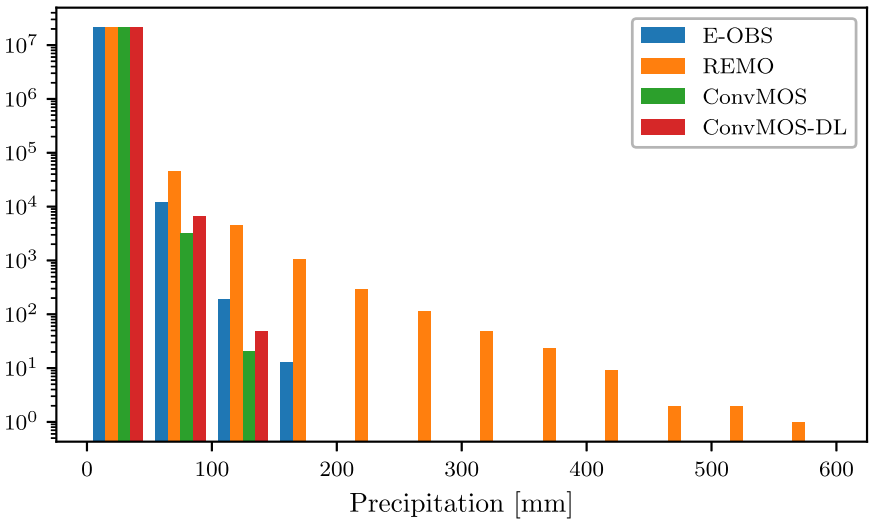


Fig. 6 Daily precipitation distribution on the test set for E-OBS's observations, the estimates of REMO, ConvMOS, and ConvMOS-DL. The 12 bins begin at 0 mm and are 50 mm wide. The y-axis (occurrences) is scaled logarithmically

Table 6 RF hyperparameter search space. "HP" stands for hyperparameter

HP	Range	HP	Range	HP	Range
n_estimators	10–2000	min_samples_split	2–10	max_depth	10–110
max_features	0.01–1.0	min_samples_leaf	1–10	bootstrap	T or F

A.1 Model training time

Applying MOS can provide more accurate climate data but it comes with an additional time burden due to the MOS's training procedure. To quantify this time burden fairly, we fit each MOS approaches five times on the same hardware—in contrast to the cluster of heterogeneous hardware used in the main experiment—and measure the training time.

All Deep Learning models (i.e. ResNets, U-Nets, CM-UNet, ConvMOS) are trained on a single Nvidia RTX 2080 TI GPU (Graphics Processing Unit), which is relatively affordable consumer hardware in comparison to expensive data center GPUs. These models are implemented in PyTorch 1.7.1 (Paszke et al. 2019) with CUDA 11.0. The other models (i.e. Lin, NL PCR, NL RF) are fitted using 15 cores of an AMD Epyc 7502P processor, which is not a standard consumer but a more expensive data center CPU (Central Processing Unit). These non-GPU models are implemented in Scikit-learn 0.23.2 (Pedregosa et al. 2011).

Table 7 shows the mean training duration in hours per MOS approach in this training duration experiment. Both NL RF and NL PCR stand out with relatively long training duration. As with Lin, NL RF and NL PCR fit one model per location, which is time

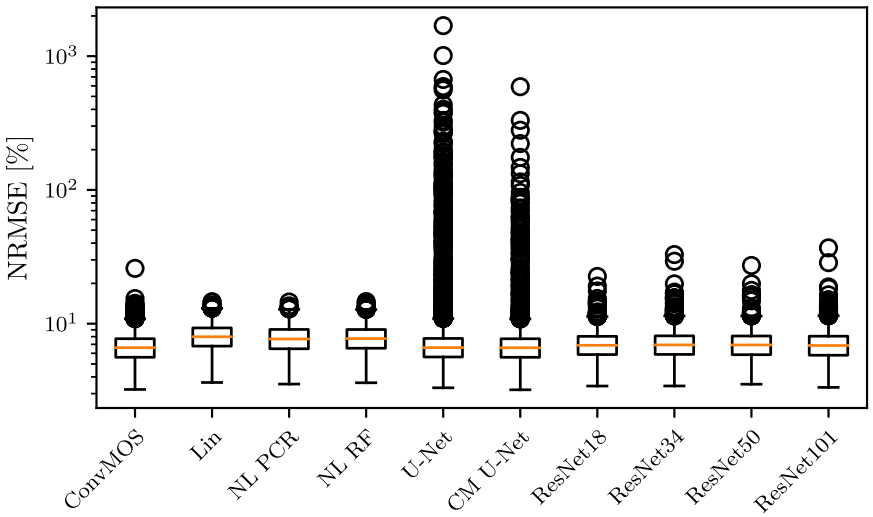


Fig. 7 Mean test NRMSEs per location. U-Net and CM U-Net show high NRMSE on mostly low-precipitation locations in contrast to all other models

Table 7 Mean training duration in hours per MOS approach

MOS	Train duration [h]	Hardware	MOS	Train duration [h]	Hardware
Lin	0.03 ± 0.00	CPU	ResNet50	0.49 ± 0.05	GPU
NL PCR	58.16 ± 25.33	CPU	ResNet101	0.62 ± 0.05	GPU
NL RF	93.32 ± 1.47	CPU	U-Net	0.42 ± 0.03	GPU
ResNet18	0.38 ± 0.03	GPU	CM-UNet	0.44 ± 0.05	GPU
ResNet34	0.49 ± 0.04	GPU	ConvMOS	1.14 ± 0.25	GPU

consuming for large study areas like the one used here with 121×121 locations. However, NL RF's and NL PCR's long training times are mostly due to supervised PCA. NL RF takes longer than NL PCR due to the higher model complexity and the per-location hyperparameter tuning, which we employ for optimal performance (see Sect. 5.3). All Deep Learning approaches are trained in under two hours. ConvMOS's training duration is comparatively long and shows high variance. Regardless, there is no large practical difference between these Deep Learning approaches with regard to training duration since all train relatively quickly. All in all, we consider these training times—except for NL PCR and NL RF—minor in comparison to the time needed for the climate simulations of the climate model, which usually takes multiple days. The Deep Learning approaches are faster in settings with large study areas while providing better performance, as seen in this work's main experiment.

A.2 Architecture composition study for CM-UNet

ConvMOS's architecture composition study shows that a combination of local and global modules is beneficial. We further confirm this and optimize CM-UNet's module composition by conducting the architecture composition study again with U-Net as the global module (CM-UNet). The experimental setup is the same except for the different global modules and the batch size of 64 instead of 128 due to GPU memory

Table 8 Mean validation metrics per CM-UNet composition sorted by RMSE (left < right), rounded to three decimal places. "Mod." is modules, "Cor." is Correlation, "g"/"l" is global/local network. RMSE, Bias in mm. NRMSE in %

Mod.	RMSE	NRMSE	Cor.	Skill	R^2	Bias	Mod.	RMSE	NRMSE	Cor.	Skill	R^2	Bias
gl	**3.264**	8.858	.751	.842	0.475	−0.009	glg	3.303	9.522	.749	.830	−0.416	0.056
ggll	3.269	8.883	.750	.836	0.461	0.019	glgg	3.306	9.777	.750	.827	−1.048	0.072
glll	3.269	**8.823**	.751	**.845**	**0.501**	0.002	lggg	3.307	10.211	.749	.827	−2.458	0.038
gglg	3.270	9.372	.751	.833	−0.290	**−0.001**	g	3.310	10.918	.753	.832	−5.774	0.040
gll	3.271	8.856	.751	.839	0.484	0.040	lgg	3.313	10.262	.749	.823	−2.590	0.013
gggl	3.272	9.029	.751	.842	0.337	0.022	gg	3.313	10.894	.753	.826	−5.479	0.039
ggl	3.272	8.978	.751	.832	0.393	0.050	lg	3.320	10.279	.747	.821	−2.679	0.037
glgl	3.274	9.029	.751	.834	0.345	0.020	llgg	3.327	10.158	.744	.821	−2.073	0.048
lgll	3.277	8.968	.749	.833	0.405	0.027	ggg	3.329	10.957	.752	.824	−5.529	0.079
lgl	3.277	9.092	.749	.837	0.255	−0.010	llg	3.339	10.102	.743	.827	−1.877	0.005
lggl	3.286	9.182	.748	.834	0.156	0.027	lllg	3.343	9.997	.741	.832	−1.532	−0.004
llgl	3.299	9.177	.745	.833	0.213	0.019	llll	3.787	10.050	.648	.795	0.375	0.188
gggg	3.301	10.888	**.754**	.830	−5.586	0.014	lll	3.789	10.053	.648	.794	0.374	0.190
lglg	3.302	9.674	.747	.824	−0.762	0.034	ll	3.793	10.063	.647	.792	0.373	0.196
gllg	3.303	9.550	.748	.829	−0.507	0.041	l	3.802	10.086	.643	.783	0.370	0.200

Bold values indicate the best value

limitations with compositions consisting of three or four U-Nets. We do not expect the change in batch size to affect the comparison considerably.

Table 8 shows validation set mean metrics for all locations with observational data of each CM-UNet architecture composition sorted by RMSE. The composition CM-UNet-gl provides the lowest RMSE as well as Correlation, Skill score, and Bias that are not significantly different to the composition with the best value in the respective metric. While it is not best in NRMSE and R^2, we choose this composition for our experiments due to its low RMSE and it being among the best compositions with regard to the other metrics. Again, we find that the combination of per-location and global model parameters can improve performance in terms of RMSE, NRMSE and R^2, where CM-UNet-gl provides significantly better performance in comparison to the best composition consisting solely of global or local modules, namely CM-UNet-gggg. Compositions without both global and local modules typically perform worse than combinations of both. An exception are Correlations, where CM-UNet-gggg performs best significantly but performs subpar especially for NRMSE and R^2. This study confirms again that a local network as the final module provides relatively good NRMSE and R^2.

A.3 Estimation quality over time

This work considers MOS where temporal climate dynamics are entirely modeled by the climate model. Daily precipitations are adjusted disregarding time. Since MOS use training data from a certain time range, it is interesting to consider error trends with increasing distance to this time period. Distributions produced by climate models may change over time, possibly leading to issues for MOS. We investigate this by analyzing the test set performance over time.

Figure 8 visualizes daily RMSE of precipitation over the test set time range for REMO, ConvMOS, and NL PCR, smoothed with a moving average window of 14

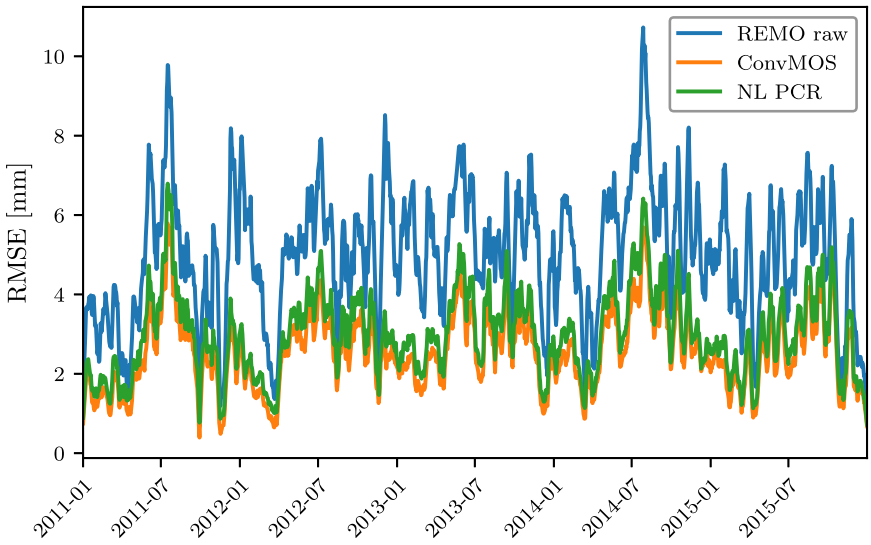


Fig. 8 RMSE of precipitation in mm for the test set across the study area over time. The graph is smoothed using a moving average window of 14 days

Table 9 Test set mean RMSE in mm per year for all locations having observational data. Values rounded to two decimal places

Year MOS	2011	2012	2013	2014	2015
REMO raw	4.93	5.20	5.45	5.72	4.86
Lin	3.41	3.32	3.55	3.80	3.31
NL PCR	3.26	3.20	3.44	3.67	3.11
NL RF	3.27 ± 0.00	3.19 ± 0.00	3.45 ± 0.00	3.70 ± 0.00	3.15 ± 0.00
ResNet18	2.93 ± 0.02	2.83 ± 0.02	3.08 ± 0.02	3.31 ± 0.01	2.85 ± 0.02
ResNet34	2.97 ± 0.02	2.87 ± 0.02	3.11 ± 0.02	3.34 ± 0.02	2.86 ± 0.02
ResNet50	2.92 ± 0.02	2.84 ± 0.01	3.08 ± 0.02	3.31 ± 0.02	2.86 ± 0.02
ResNet101	2.92 ± 0.03	2.84 ± 0.02	3.07 ± 0.02	3.32 ± 0.02	2.84 ± 0.02
U-Net	2.87 ± 0.03	2.78 ± 0.02	3.00 ± 0.02	3.21 ± 0.02	2.80 ± 0.02
CM-UNet	**2.82 ± 0.02**	**2.73 ± 0.02**	**2.96 ± 0.02**	**3.17 ± 0.01**	**2.76 ± 0.02**
ConvMOS	**2.82 ± 0.02**	2.75 ± 0.01	**2.96 ± 0.02**	3.18 ± 0.02	**2.76 ± 0.02**

Bold values indicate the best value

Table 10 Test set mean RMSE relative to REMO's RMSE in % per year for all locations having observational data. Values rounded to percentages

Year MOS	2011	2012	2013	2014	2015
REMO raw	100	100	100	100	100
Lin	69	64	65	67	68
NL PCR	66	62	63	64	64
NL RF	66 ± 0	61 ± 0	63 ± 0	65 ± 0	66 ± 0
ResNet18	59 ± 0	55 ± 0	57 ± 0	58 ± 0	59 ± 0
ResNet34	60 ± 0	55 ± 0	57 ± 0	58 ± 0	59 ± 0
ResNet50	59 ± 0	55 ± 0	57 ± 0	58 ± 0	59 ± 0
ResNet101	59 ± 1	55 ± 0	56 ± 0	58 ± 0	58 ± 0
U-Net	58 ± 1	**53 ± 0**	55 ± 0	56 ± 0	58 ± 1
CM-UNet	**57 ± 0**	**53 ± 0**	**54 ± 0**	**55 ± 0**	**57 ± 0**
ConvMOS	**57 ± 0**	**53 ± 0**	**54 ± 0**	56 ± 0	**57 ± 0**

Bold values indicate the best value

days to reduce noise. It shows that MOS RMSE mostly follows REMO's RMSE but on a lower level. We find no noticeable trend in RMSE for the models depicted, as well as the other MOS approaches. We come to the same conclusion when considering Table 9, which shows the absolute RMSE per year for all MOS approaches and REMO, and Table 10, which shows the relative RMSE per year for each MOS approach as percentages of REMO's RMSE. Especially the latter table shows for all MOS techniques only small RMSE fluctuations of at the very most 5 % relative to REMO's RMSE, suggesting that MOS error trends follow REMO's error trends.

While the limited timespan available does not allow for a conclusive answer regarding error trends for longer timespans, the data does suggest that time difference between training data and test data may have no or only a minor influence on errors. Neverthe-

less, when considering longer timespans than five years, the climate model output's distribution may change to such an extent, that there may be a noticeable effect. We suggest to analyze this in climate studies that apply MOS in order to detect potential issues with distributional shifts.

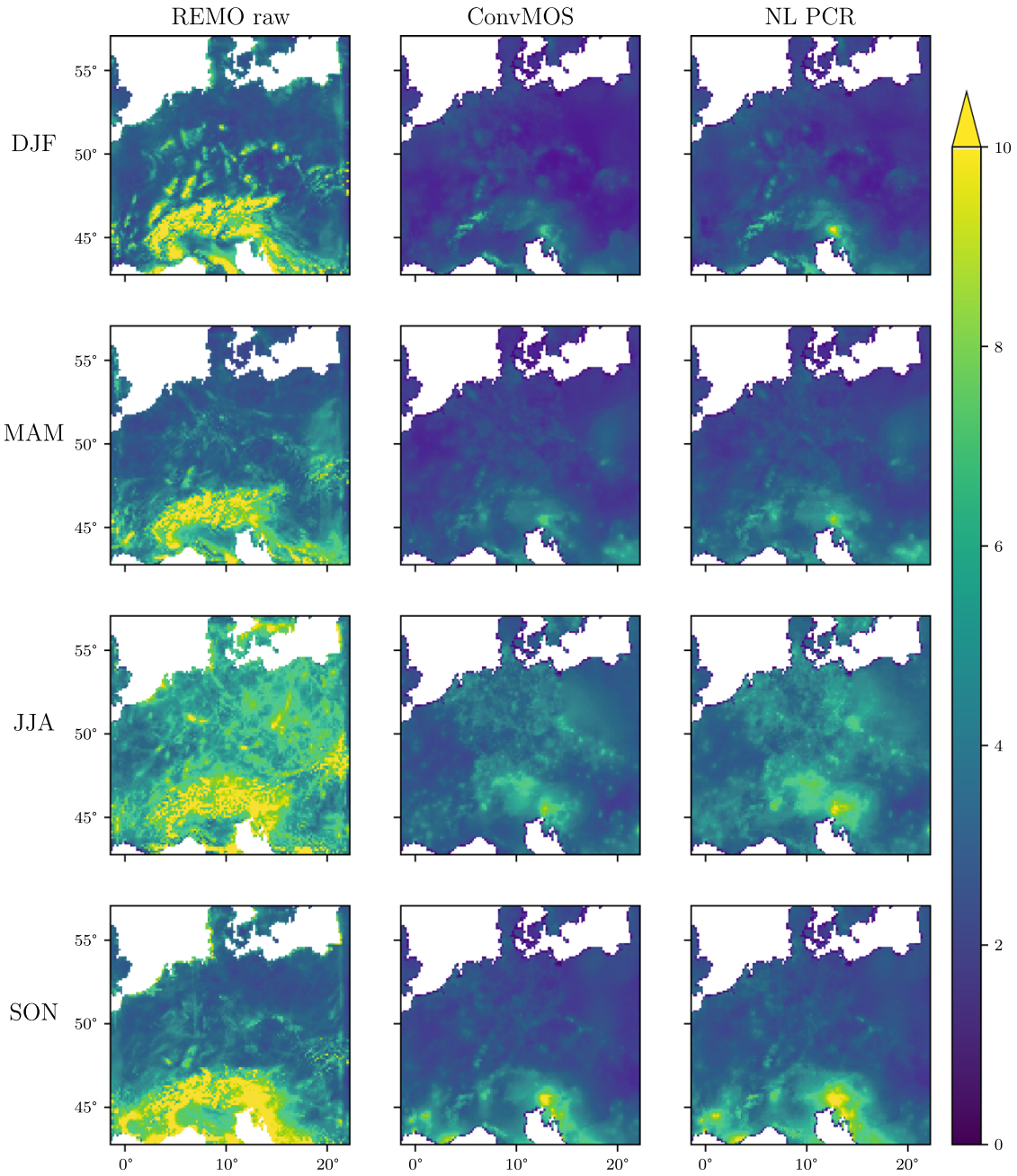


Fig. 9 Test RMSE in mm per location and season (DJF is December, January, and February; MAM is March, April, and May; JJA is June, July, and August; SON is September, October, and November). REMO has larger RMSEs than 10 mm but the colorbar's extent is limited to better show general performance

A.4 Seasonal results over the study area

Seasonal results are visualized in Fig. 9 for REMO's raw output, ConvMOS, and NL PCR, which is overall the best standard MOS approach in terms of RMSE. We show the same model instances as in Fig. 4. During all seasons, we find REMO's largest errors in mountainous regions like the Alps. This also shows in the RMSE of both ConvMOS's and NL PCR's output where these areas often continue to have more pronounced errors. The season with the largest error JJA shows more evenly distributed large RMSE values across the study area compared to the other seasons, resulting also in comparatively large RMSE in the MOS outputs. The relatively large overall RMSE values of season SON concentrate in the Alps and the Mediterranean coast, while RMSE for cells north of the Alps seem similar to those during seasons DJF and MAM. Matching the findings of Table 4, we tend to see lower RMSE with ConvMOS in comparison to NL PCR. For example, the latter has more difficulties in reducing the large errors near the border between Italy and Slovenia and we also often see slightly larger RMSEs north of the Alps in comparison to our approach. These results show that ConvMOS can be better than standard MOS approaches at improving precipitation estimates regardless of the season.

References

Abdar M et al (2021) A review of uncertainty quantification in deep learning: techniques, applications and challenges. In: Information fusion

Agrawal S, Barrington L, Bromberg C, Burge J, Gazen C, Hickey J (2019) Machine learning for precipitation nowcasting from radar images. arXiv:1912.12132

Ahmed K, Shahid S, Nawaz N, Khan N (2019) Modeling climate change impacts on precipitation in arid regions of Pakistan: a non-local model output statistics downscaling approach. Theor Appl Climatol 137:1–2

Bair E, Hastie T, Paul D, Tibshirani R (2006) Prediction by supervised principal components. In: JASA 101.473, pp 119–137

Berrisford P et al (2011). The ERA-interim archive. Version 2.0. In: ECMWF

Caruana R, Lawrence S, Giles CL (2001) Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping. In: NeurIPS

Cavalcante L, Bessa RJ, Reis M, Browell J (2017) LASSO vector autoregression structures for very short-term wind power forecasting. Wind Energy 20(4):657–675

Ceci M, Corizzo R, Malerba D, Rashkovska A (2019) Spatial autocorrelation and entropy for renewable energy forecasting. Data Min Knowl Discov 33(3):698–729

Chen M-J, Lin C-Y, Wu Y-T, Wu P-C, Lung S-C, Su H-J (2012) Effects of extreme precipitation to the distribution of infectious diseases in Taiwan, 1994–2008. PLoS ONE 7(6):e34651

Corizzo R, Ceci M, Fanaee-T H, Gama J (2021) Multi-aspect renewable energy forecasting. Inf Sci 546:701–722

Cornes RC, van der Schrier G, van den Besselaar EJ, Jones PD (2018) An ensemble version of the E-OBS temperature and precipitation data sets. J Geophys Res Atmosp 123(17):9391–9409

DAAC, EDC (1996) GTOPO 30 Database. In: US Geological Survey

Dee D et al (2011) The ERA-Interim reanalysis: configuration and performance of the data assimilation system. Q J R Meteorol Soc 137:553–597

Deutscher Wetterdienst (2021) Warnkriterien. https://www.dwd.de/DE/wetter/warnungen_aktuell/kriterien/warnkriterien.html

Eden JM, Widmann M (2014) Downscaling of GCM-simulated precipitation using model output statistics. JCLI 27(1):312–324

Gesch DB, Verdin KL, Greenlee SK (1999) New land surface digital elevation model covers the earth. Eos 80(6):69–70. https://doi.org/10.1029/99EO00050
Gonçalves C, Cavalcante L, Brito M, Bessa RJ, Gama J (2021) Forecasting conditional extreme quantiles for wind energy. Electr Power Syst Res 190:106636
Grönquist P et al (2021) Deep learning for post-processing ensemble weather forecasts. Philos Trans R Soc A 379(2194):20200092
Hagemann S (2002). An improved land surface parameter dataset for global and regional climate models. https://doi.org/10.17617/2.2344576
Haylock MR, Hofstra N, Klein Tank AMG, Klok EJ, Jones PD, New M (2008) A European daily high-resolution gridded data set of surface temperature and precipitation for 1950–2006. JGR Atmosp 113(20):D20119
He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. CVPR 2016:770–778
He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
Jacob D (2001) A note to the simulation of the annual and inter-annual variability of the water budget over the Baltic Sea drainage basin. Meteorol Atmosp Phys 77(1–4):61–73
Jacob D et al (2001) A comprehensive model inter-comparison study investigating the water budget during the BALTEX-PIDCAP period. Meteorol Atmosp Phys 77(1–4):19–43
Kandel ME et al (2020) Phase imaging with computational specificity (PICS) for measuring dry mass changes in sub-cellular compartments. Nat Commun 11(1):1–10
Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
Kotlarski S (2007) A Subgrid Glacier Parameterisation for Use in Regional Climate Modelling. Ph.D. thesis. Hamburg, p 178
Kundzewicz ZW (2003) Extreme precipitation and floods in the changing world. IAHS Publ 281:32–39
LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. In: IEEE 86.11, pp 2278–2324
Liu Y, Ganguly AR, Dy J (2020) Climate downscaling using YNet: a deep convolutional network with skip connections and fusion. In: KDD 2020
Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
Lüthi D, Heinzeller D (2017) Leitfaden zur Nutzung dynamischer regionaler Klimamodelle. In: promet 99, p 49
Majewski D (1991) The Europa-modell of the Deutscher Wetterdienst. In: ECMWF "numerical methods in atmospheric models" 2, pp 147–191
Moghim S, Bras RL (2017) Bias correction of climate modeled temperature and precipitation using artificial neural networks. J Hydrometeorol 18:1867–1884
Nair V, Hinton GE (2010) Rectified linear units improve restricted Boltzmann machines. In: ICML
Noor M, Ismail T bin, Ullah S, Iqbal Z, Nawaz N, Ahmed K (2019) A non-local model output statistics approach for the downscaling of CMIP5 GCMs for the projection of rainfall in Peninsular Malaysia. In: JWCC
Paeth H (2011) Postprocessing of simulated precipitation for impact research in West Africa. Part I: model output statistics for monthly data. Clim Dyn 36(7–8):1321–1336
Paszke A et al (2019) PyTorch: an imperative style, high-performance deep learning library. In: NeurIPS. Curran Associates, Inc., pp 8024–8035
Pedregosa F et al (2011) Scikit-learn: machine learning in python. In: JMLR
Perkins S, Pitman A, Holbrook N, McAneney J (2007) Evaluation of the AR4 climate models' simulated daily maximum temperature, minimum temperature, and precipitation over Australia using probability density functions. J Clim 20(17):4356–4376
Pour SH, Shahid S, Chung E-S, Wang X-J (2018) Model output statistics downscaling using support vector machine for the projection of spatial and temporal changes in rainfall of Bangladesh. Atmos Res 213:149–162
Qin Z, Cen C, Guo X (2019) Prediction of air quality based on KNN-LSTM. J Phys Conf Ser 1237:4
Rasp S, Pritchard MS, Gentine P (2018) Deep learning to represent subgrid processes in climate models. PNAS 115(39):9684–9689

Roeckner E et al (1996) The atmospheric general circulation model ECHAM4: model description and simulation of present-day climate. Max-Planck-Institute of Meteorology, Technical report Hamburg, p 171

Rolnick D et al (2022) Tackling climate change with machine learning. ACM Comput Surv. https://doi.org/10.1145/3485128

Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: MICCAI. Springer, pp 234–241

Sa'adi Z, Shahid S, Chung E-S, Ismail T (2017) Projection of spatial and temporal changes of rainfall in Sarawak of Borneo Island using statistical downscaling of CMIP5 models. Atmos Res 197:446–460

Samuelsson P, Kourzeneva E, Mironov D (2010) The impact of lakes on the European climate as simulated by a regional climate model. Boreal Environ Res 15:113–129

Schulzweida U (2019). CDO. https://doi.org/10.5281/zenodo.3539275

Semmler T (2002) Der Wasser- und Energiehaushalt der arktischen Atmosphre. PhD thesis. Hamburg, pp 1–123

Shi X et al (2017) Deep learning for precipitation nowcasting: a benchmark and a new model. In: NeurIPS, pp 5617–5627

Shi G, Leung Y, Zhang JS, Fung T, Du F, Zhou Y (2021) A novel method for identifying hotspots and forecasting air quality through an adaptive utilization of spatio-temporal information of multiple factors. Sci Total Environ 759:143513

Silver NC, Dunlap WP (1987) Averaging correlation coefficients: should Fisher's z transformation be used? J Appl Psychol 72(1):146

Steininger M, Abel D, Ziegler K, Krause A, Paeth H, Hotho A (2020) Deep learning for climate model output statistics. arXiv:2012.10394

Steininger M, Kobs K, Davidson P, Krause A, Hotho A (2021) Density-based weighting for imbalanced regression. Mach Learn 110(8):2187–2211

Teichmann C (2010) Climate and air pollution modelling in south America with focus on megacities. Ph.D. thesis. Hamburg, p 167

Vandal T, Kodra E, Ganguly S, Michaelis A, Nemani R, Ganguly AR (2017) Deepsd: generating high resolution climate change projections through single image super-resolution. KDD 2017:1663–1672

Wilcoxon F (1945) Individual comparisons by ranking methods. Biomet Bull 1(6):80–83

Zeppel M, Wilks JV, Lewis JD (2014) Impacts of extreme precipitation and seasonal changes in precipitation on plants. Biogeosciences 11:11

Zhang Q, Lam JC, Li VO, Han Y (2020) Deep-AIR: a hybrid CNN-LSTM framework for Fine-grained air pollution forecast. arXiv:2001.11957

Machine Learning
https://doi.org/10.1007/s10994-021-06023-5

Density-based weighting for imbalanced regression

Michael Steininger[1] · **Konstantin Kobs**[1] · **Padraig Davidson**[1] · **Anna Krause**[1] · **Andreas Hotho**[1]

Received: 22 November 2020 / Revised: 23 April 2021 / Accepted: 16 June 2021

Abstract

In many real world settings, imbalanced data impedes model performance of learning algorithms, like neural networks, mostly for rare cases. This is especially problematic for tasks focusing on these rare occurrences. For example, when estimating precipitation, extreme rainfall events are scarce but important considering their potential consequences. While there are numerous well studied solutions for classification settings, most of them cannot be applied to regression easily. Of the few solutions for regression tasks, barely any have explored cost-sensitive learning which is known to have advantages compared to sampling-based methods in classification tasks. In this work, we propose a sample weighting approach for imbalanced regression datasets called *DenseWeight* and a cost-sensitive learning approach for neural network regression with imbalanced data called *DenseLoss* based on our weighting scheme. DenseWeight weights data points according to their target value rarities through kernel density estimation (KDE). DenseLoss adjusts each data point's influence on the loss according to DenseWeight, giving rare data points more influence on model training compared to common data points. We show on multiple differently distributed datasets that DenseLoss significantly improves model performance for rare data points through its density-based weighting scheme. Additionally, we compare DenseLoss to the state-of-the-art method SMOGN, finding that our method mostly yields better performance. Our approach provides more control over model training as it enables us to actively decide on the trade-off between focusing on common or rare cases through a single hyperparameter, allowing the training of better models for rare data points.

Keywords Imbalanced regression · Cost-sensitive learning · Sample weighting · Kernel-density estimation · Supervised learning

Editors: Annalisa Appice, Sergio Escalera, Jose A. Gamez, Heike Trautmann.

✉ Michael Steininger
steininger@informatik.uni-wuerzburg.de

Extended author information available on the last page of the article

1 Introduction

Many machine learning algorithms, like neural networks, typically expect roughly uniform target distributions (Cui et al. 2019; Krawczyk 2016; Sun et al. 2009). In the case of classification that means that there are similar numbers of examples per class. For regression there should be a similar density of samples across the complete target value range. However, many datasets exhibit skewed target distributions with target values in certain ranges occurring less frequently than others. Consequently, models can become biased, leading to better performance for common cases than for rare cases (Cui et al. 2019; Krawczyk 2016). This is particularly problematic for tasks where these rare occurrences are of special interest. Examples include precipitation estimation, where extreme rainfall is rare but can have dramatic consequences, or fraud detection, where rare fraudulent events are supposed to be detected.

There are many solutions to this problem for classification tasks including resampling strategies (Chawla et al. 2002; He et al. 2008) and cost-sensitive learning approaches (Cui et al. 2019; Huang et al. 2016; Wang et al. 2017). However, these cannot be applied easily to regression tasks because of the inherent differences between continuous and discrete, nominal target values. Typical solutions to data imbalance require a notion of rarity or importance for a data point in order to know which data points to over- and undersample or which data points to weight more strongly. It is harder to define which values are rare for regression tasks in comparison to classification tasks, since one cannot simply use class frequencies (Branco et al. 2017). Only few works explore methods improving model performance for rare cases in regression settings, mostly proposing sampling-based approaches (Branco et al. 2017; Krawczyk 2016; Torgo et al. 2013). These can have disadvantages in comparison to cost-sensitive methods since the creation of new data points via oversampling of existing data points may lead to overfitting as well as additional noise, while undersampling removes information (Cui et al. 2019; Dong et al. 2017). The success of cost-sensitive learning for imbalanced classification tasks suggests that exploring this direction for imbalanced regression could also lead to better methods in this domain (Krawczyk 2016).

In this paper, we propose a sample weighting approach for imbalanced regression datasets called DenseWeight and, based on this, a cost-sensitive learning method for imbalanced regression with neural networks called DenseLoss. Our approach is visualized in Fig. 1: (i) We approximate the density function of the training target values using KDE. (ii) The resulting density function forms the basis for calculating DenseWeight's weighting function. (iii) DenseLoss assigns each data point in the training set a weight according to DenseWeight, increasing the influence of rare data points on the loss and the gradients. We introduce a single, easily interpretable hyperparameter, which allows us to configure to which extent we shift a model's focus towards rare regions of the target variable's distribution.

Our contributions are as follows: (i) We propose DenseWeight, a sample weighting approach for regression with imbalanced data. (ii) We propose DenseLoss, a cost-sensitive learning approach based on DenseWeight for neural network regression models with imbalanced data. (iii) We analyze DenseLoss 's influence on performance for common and rare data points using synthetic data. (iv) We compare DenseLoss to the state-of-the-art imbalanced regression method SMOGN, finding that our method typically provides better performance. (v) We apply DenseLoss to the heavily imbalanced

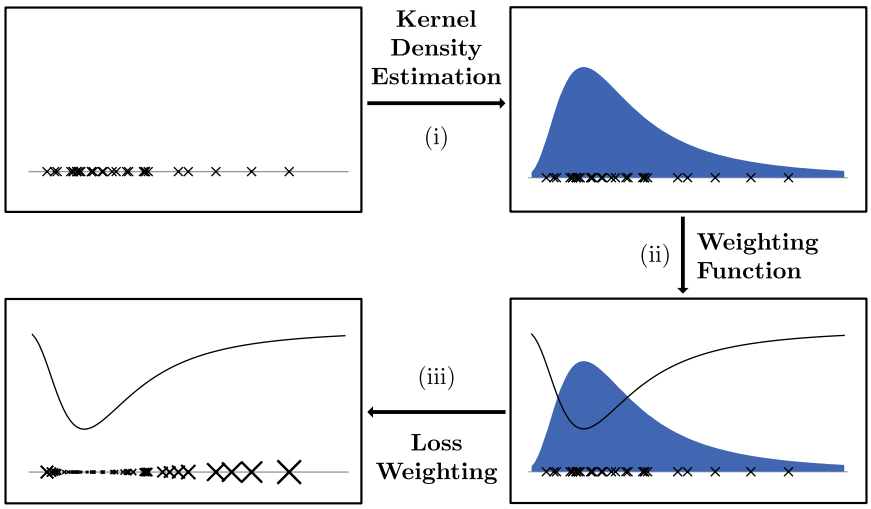


Fig. 1 Given the target values of all training examples, we (*i*) compute a kernel density estimation (KDE) that approximates the target value distribution, (*ii*) calculate a weighting function from the resulting probability density function, and (*iii*) weight the loss for each data point in the training procedure

real world problem of downscaling precipitation, showing that it is able to significantly improve model performance in practice.

2 Related work

Imbalanced data can in principle be tackled with data-level methods, algorithm-level methods, or a combination of both (Krawczyk 2016). Data-level methods typically over- and/or undersample subsets of a dataset to balance the distribution. Algorithm-level methods modify existing learning algorithms to better cope with imbalanced data.

There are many solutions to data imbalance for classification tasks. Data-level methods for classification often create new samples for rare classes (oversampling) and/or remove samples of common classes (undersampling). Notable examples include ADASYN (He et al. 2008) and SMOTE (Chawla et al. 2002). Recently, KDE was used to estimate the feature distribution of minority classes (Kamalov 2020). New minority class samples are generated using the estimated feature distribution. In contrast to Kamalov (i) we use KDE to measure rarity on a continuous target domain and not to model features, (ii) we do not generate samples, and (iii) we devise our method for regression. Algorithm-level methods for classification typically involve cost-sensitive learning, where the loss of samples with rare classes is emphasized in the overall loss (Cui et al. 2019). Weighting is often based on the inverse class frequency as a measure of rarity (Huang et al. 2016; Wang et al. 2017). We propose a conceptually similar method, but for regression instead of classification. The continuous target variable of regression tasks makes it harder to determine a single sample's rarity, preventing simple adaptations of existing cost-sensitive learning approaches (Branco et al. 2017).

While there is work on cost-sensitive learning for regression models, these approaches assign different costs to over- and underestimation respectively, regardless of a data point's rarity (Zhao et al. 2011; Hernández-Orallo 2013). However, we are interested in exploring how cost-sensitive learning can be used to solve the problem of imbalanced datasets for regression tasks, for which only few works exist. There is a cost-sensitive post-processing technique called probabilistic reframing which adjusts

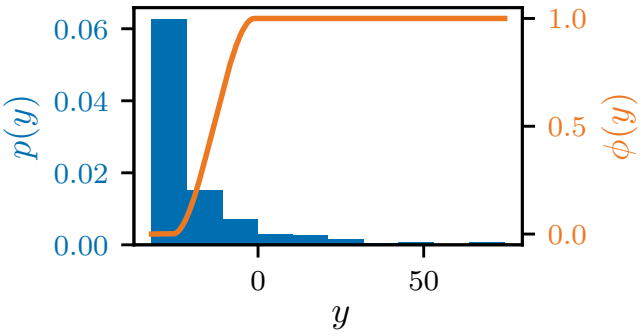


Fig. 2 SmoteR and SMOGN's relevance function ϕ for pareto-distributed data

estimates of previously built models to different contexts (Hernández-Orallo 2014). It would be feasible to apply this to imbalanced domains but it was not evaluated for this yet (Branco et al. 2016b). A cost-sensitive method for obtaining regression tree ensembles biased according to a utility function is ubaRules (Ribeiro 2011) which is mostly used to estimate extreme values as accurately as possible. It is specific to regression tree ensembles while our proposal is designed for—but not restricted to—the use with neural networks. A metric that takes both rare, extreme samples and common samples into account for evaluating a model's ability to predict extreme values is SERA (Ribeiro and Moniz 2020). SERA can be considered a loss function that is used for model selection and hyperparameter optimization but it is not incorporated in a learning method like DenseLoss.

Despite the lack of cost-sensitive approaches, there are sampling-based data-level methods which are applied during data pre-processing. One approach is SMOTE for regression (SmoteR) (Torgo et al. 2013), which is based on the original SMOTE method for classification (Chawla et al. 2002). It combines undersampling of common data points and oversampling of rare cases, in order to create a more balanced distribution. The authors adjust SMOTE to work for regression domains by binning data points into relevant and irrelevant partitions using a relevance threshold t_R and a relevance function ϕ. They use an automatic method for obtaining ϕ based on box plot statistics through which specific control points on the target domain are obtained. Each control point is a tuple $(y, \phi(y), \phi'(y))$, where $\phi'(y)$ —the derivative of relevance $\phi(y)$—is always set to 0, since control points are assumed to be local extrema of relevance. The relevance function ϕ is then defined with piecewise cubic Hermite interpolation through these control points (Ribeiro 2011). Figure 2 shows a resulting ϕ for data following a pareto distribution. This automatic method for obtaining ϕ assumes that extreme values are rare, which is in contrast to our work, where rare values are automatically detected without such assumptions. Data points marked as relevant ($\phi(y) > t_R$) are oversampled, creating new synthetic cases via interpolation of features and target values between two relevant data points. Irrelevant data points are undersampled.

The SMOGN (Branco et al. 2017) algorithm builds on SmoteR and combines it with oversampling via Gaussian noise. For the latter, normally distributed noise is added to the features and the target value of rare data points, creating additional, slightly altered replicas of existing samples (Branco et al. 2016a). Rare data points are identified using the same method for obtaining a relevance function ϕ used by SmoteR. SMOGN iterates over all rare samples and selects between SmoteR's interpolation based oversampling and Gaussian noise based oversampling depending on the distance to the k-nearest neighbors. For small distances, SmoteR's interpolation is applied, since interpolation is deemed more reliable for close samples. Other rare data points are oversampled with Gaussian noise. Common data points are randomly undersampled. The authors report improvements compared to

SmoteR (Branco et al. 2017). Because of this and a lack of other methods, SMOGN can be considered the state-of-the-art.

In contrast to these data-level methods, we propose an algorithm-level, cost-sensitive method for imbalanced regression called DenseLoss using our density-based weighting scheme DenseWeight. The concept of weighting data points based on the target value distribution is already present in prior work, e.g. in the automatic method for obtaining relevance functions used by SMOGN, or in SERA. However, DenseWeight does not make assumptions about which cases are rare since it determines relative rarity with a density function. Contrary to SmoteR and SMOGN, DenseLoss does not explicitly change the dataset, e.g. by creating new samples.

3 Method

In this section we introduce DenseWeight, our proposed sample weighting approach for imbalanced datasets in regression tasks, and DenseLoss, our cost-sensitive learning approach for imbalanced regression problems based on DenseWeight.

3.1 DenseWeight

Our goal is to weight individual data points based on the rarity of their target values. Thus, we want to calculate a weight for each sample inversely proportional to the probability of the target value's occurrence. This is similar to the relevance functions used by the resampling approach SMOGN but we base our weighting directly on the target distribution's density function instead of box plot statistics (Branco et al. 2017). We call our density-based weighting scheme DenseWeight. We design its weighting function f_w so that the degree of weighting can be controlled by a hyperparameter $\alpha \in [0, \infty)$ with the following properties.

P.1 Samples with more common target values get smaller weights than rarer samples.
P.2 f_w yields uniform weights for $\alpha = 0$, while larger α values further emphasize the weighting scheme. This provides intuition for the effects of α.
P.3 No data points are weighted negatively, as models would try to maximize the difference between estimate and true value for these data points during training.
P.4 No weight should be 0 to avoid models ignoring parts of the dataset.
P.5 The mean weight over all data points is 1. This eases applicability for model optimization with gradient descent as it avoids influence on learning rates.

These weights can theoretically be applied to any type of machine learning model that allows for sample weighting to allow fitting models better suited for the estimation of rare cases. We will use them for our cost-sensitive imbalanced regression approach for neural networks DenseLoss in this work. Next, we define how the rarity of a data point is measured, before designing the weighting function f_w with these properties.

3.1.1 Measure of rarity

In order to weight data points based on the rarity of their target values, we need a measure of rarity for f_w. To this end we want to determine the target variable's density function p. Values of density functions can be interpreted as relative measures of density, allowing the

distinction between rare and common value ranges (Grinstead and Snell 2012). To obtain density function p for a dataset with N data points and target values $Y = \{y_1, y_2, \ldots, y_N\}$, we approximate it with KDE, which is a non-parametric approach to estimating a density function (Silverman 1986):

$$p(y) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{y - y_i}{h}\right) \tag{1}$$

with kernel function K and bandwidth h. Literature shows that the choice of kernel function is rather unimportant for KDE with only small differences between common kernel functions (Chen 2017), which is why we use Gaussian kernels. For bandwidth selection, we found that, in practice, the automatic bandwidth selection method Silverman's rule (Silverman 1986) produces density functions which follow the distributions well for the datasets used in this work. KDE allows calculating a density value per data point. Since it does not affect relative density information, we can normalize all data points' density values in the training set to a range between 0 and 1:

$$p'(y) = \frac{p(y) - min(p(Y))}{max(p(Y)) - min(p(Y))}, \tag{2}$$

where $p(Y)$ is the element-wise application of p to Y.

This normalized density function $p' \in [0, 1]$ provides intuitively interpretable values. For example, the data point in the most densely populated part of Y is assigned a value of 1, while the data point in the most sparsely populated part of Y is assigned a value of 0. Note that this normalization does not work for completely uniform data but there is no reason to apply DenseWeight with uniformly distributed data anyways.

3.1.2 Weighting function

In this section, we introduce DenseWeight 's final weighting function f_w in a step wise manner. To this end, we use the normalized density function p', hyperparameter α, and a small, positive, real-valued constant ϵ. Initially, we define a basic weighting function:

$$f'_w(\alpha, y) = 1 - \alpha p'(y). \tag{3}$$

This function already satisfies properties **P.1** and **P.2**, since $-p'$ yields larger values for rare data points compared to more common data points and α scales p', controlling the strength of density-based weighting. Setting $\alpha = 0$ has the intuitive effect of disabling density-based weighting, while $\alpha = 1$ leads to the most common data point's weight reaching 0 in this basic weighting function. Accordingly, all weights are positive for $\alpha < 1$, while $\alpha > 1$ leads to negative weights for the most common data points. The defined behavior of the α values 0 and 1 provides intuition for the choice of sensible values. However, there are still desired properties which f'_w does not satisfy. For example, we want to avoid negative and 0 weights as described in properties **P.3** and **P.4**. To this end, we clip f'_w at the small, positive, real-valued constant ϵ:

$$f''_w(\alpha, y) = max(1 - \alpha p'(y), \epsilon). \tag{4}$$

Function f''_w satisfies all desired properties except for **P.5**. Using it for weighting a cost-sensitive model optimization approach based on gradient descent like DenseLoss would

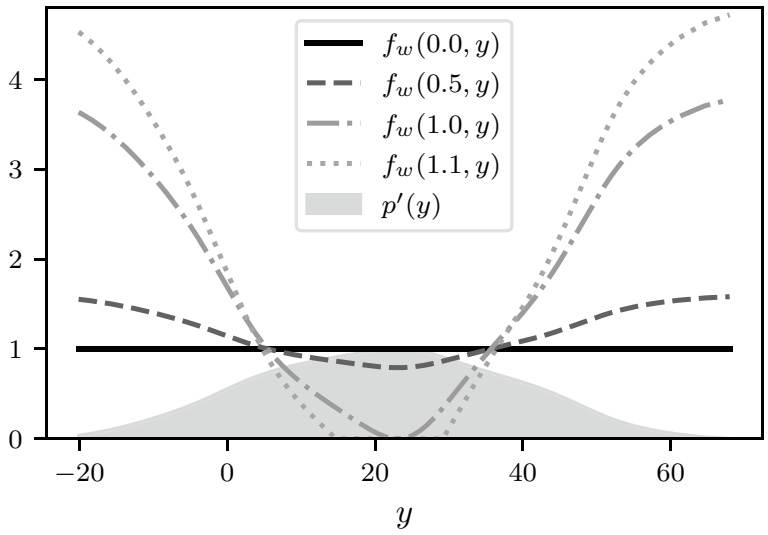


Fig. 3 DenseWeight for data sampled from a Gaussian distribution. With $\alpha = 0$ each sample's weight is 1. Higher α stretches the function, emphasizing density differences. For $\alpha > 1$ (neglecting ϵ) the function is partly clipped to avoid negative weights

influence the learning rate since α is scaling all gradients without any normalization. Changing α would also require a different learning rate if the magnitude of model parameter changes is to stay consistent. Finding a sensible learning rate would be tedious. Dividing f''_w by its mean value over all data points of the training set corrects this. The mean weight becomes 1, preventing a change in the average gradients magnitude. This leads us to DenseWeight's weighting function f_w:

$$f_w(\alpha, y) = \frac{f''_w(\alpha, y)}{\frac{1}{N}\sum_{i=1}^{N} f''_w(\alpha, y_i)} = \frac{max(1 - \alpha p'(y), \epsilon)}{\frac{1}{N}\sum_{i=1}^{N}(max(1 - \alpha p'(y_i), \epsilon))}. \tag{5}$$

Figure 3 visualizes DenseWeight for a Gaussian distributed target variable. With increasing α, weight differences between common and rare data points are emphasized more strongly. Setting $\alpha = 1$ yields a weighting function that barely reaches ϵ for the most common data points. To push more of the common data points towards a weight of ϵ, α can be increased beyond 1.

The most suitable α value for a specific task can be found by conducting a hyperparameter study. DenseLoss 's α allows for easy adjustment of the trade-off between focusing on common or rare parts of a dataset. Thus, there needs to be a definition (at least implicitly) for the meaning of performance regarding the task at hand, making it impossible to give a general rule for an optimal α.

3.2 DenseLoss

In this work we focus on neural networks due to their broad applicability to both simple and complex regression problems through the use of either relatively small multilayer perceptrons (MLPs) or large deep learning neural networks, respectively. Neural networks are typically optimized with gradient descent optimization algorithms that, given model estimates $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N\}$, aim to minimize a metric M that is incorporated into a loss function L for which we can apply sample weighting. When combining DenseWeight and sample weighting for loss functions we obtain a cost-sensitive approach for regression with imbalanced datasets, which we call DenseLoss:

$$L_{DenseLoss}(\alpha) = \frac{1}{N}\sum_{i=1}^{N} f_w(\alpha, y_i) \cdot M(\hat{y}_i, y_i)\,. \tag{6}$$

Weighting the loss per sample with DenseWeight affects the gradients' magnitude calculated based on each sample. Rarer samples yield larger gradients than more common samples even when the model's estimates are equally good according to the chosen metric. Thus, the gradients focus more on achieving best possible estimates for rare samples than for common samples. When updating model parameters with these gradients, this leads to models better suited for estimating rare samples. Similarly to cost-sensitive imbalanced classification methods weighting samples according to the inverse class frequency (Cui et al. 2019), DenseLoss is also cost-sensitive as it adapts the cost for rare samples in comparison to common samples according to the weights assigned by DenseWeight. In contrast to SMOGN, the state-of-the-art method for imbalanced regression, our approach works at the algorithm-level instead of the data-level. Weighting a loss function with DenseWeight is a very flexible approach in principle as it allows for optimization using any gradient descent optimization algorithm and any metric. Models trained with DenseLoss are expected to typically perform better for rare cases compared to models trained with uniform sample weights, as we show next.

4 Experiments

We evaluate DenseWeight and DenseLoss with three experiments: a case study on synthetic data, a comparison to the state-of-the-art, and an application to a real world task. First, we examine with synthetic datasets how DenseLoss behaves for different α values and different distribution characteristics, validating that DenseLoss is working as designed. Second, we compare DenseLoss to the state-of-the-art imbalanced regression method SMOGN, showing that our algorithm-level method can typically provide better performance for rare data points than SMOGN's data-level approach. Finally, we apply DenseLoss to the real world task statistical downscaling of precipitation, proving that it can also work for larger datasets and more complex neural network architectures.

For all experiments, we use the library KDEpy's convolution-based KDE implementation FFTKDE. It provides fast density estimation that can, however, only be evaluated on an equidistant grid (Odland 2019). Thus, for each training dataset we span a grid over the target range and assign each data point the density of the closest grid point. We use an equidistant grid with 4096 points, which is 4 times KDEpy's default resolution, to avoid potential negative effects on our method due to low KDE accuracy. In general, the quality of the resulting density function with respect to the real target distribution can be limited by low quality training data with noisy outliers. While we did not encounter such problems in this work, careful data cleaning and tuning of the KDE may improve this for such datasets. To provide a small, positive value to DenseLoss 's clipping constant ϵ we set it to 10^{-6} for all experiments. When we report significantly different results for the experiments, the statistical significance is calculated for the metrics on test datasets with the Wilcoxon signed-rank test (Wilcoxon 1945) and a significance level of 0.05. Our experiments' code and data is available[1].

[1] https://github.com/SteiMi/density-based-weighting-for-imbalanced-regression

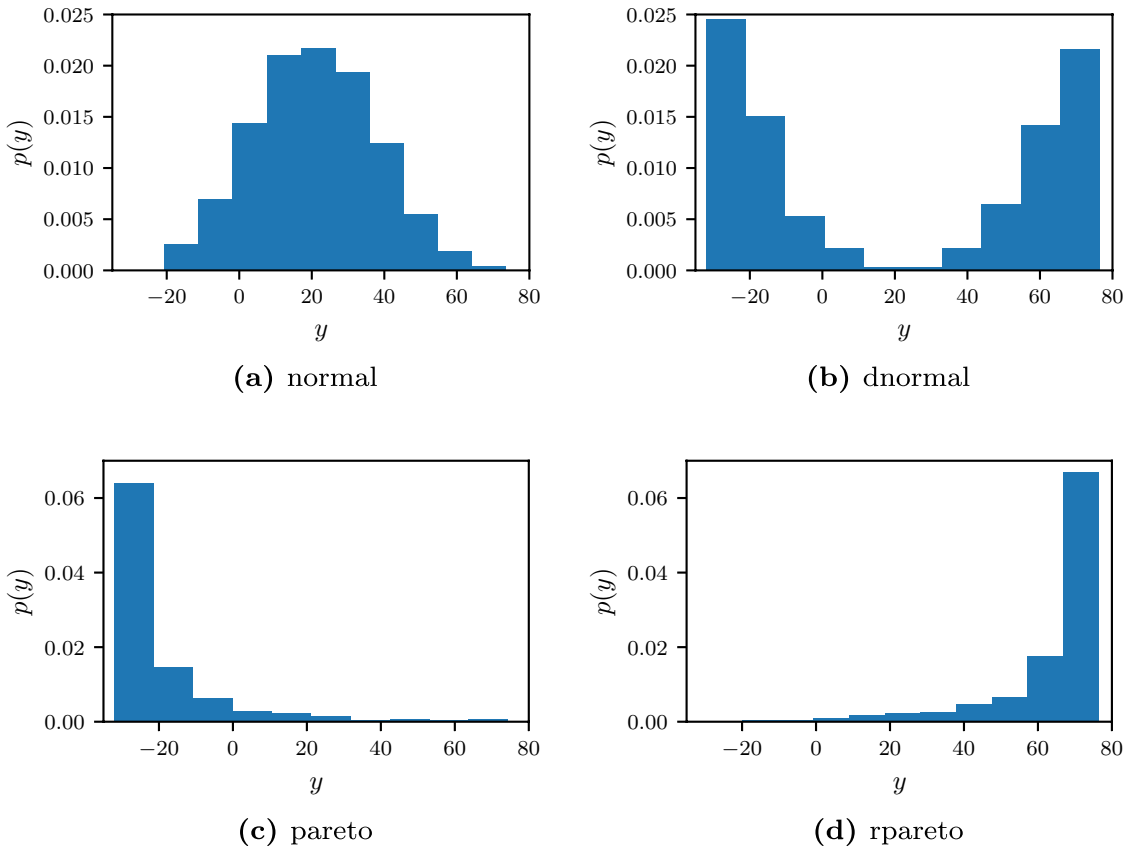


Fig. 4 Distribution of the target variable for each synthetic dataset

4.1 Case study with synthetic data

In this case study, we aim to validate the expectation that models trained with DenseLoss achieve improved performance in underrepresented parts of the dataset compared to a regular training procedure. To this end, we use four synthetic datasets with varying characteristics: two heavy-tailed datasets, following a pareto (*pareto*) and a reversed pareto distribution (*rpareto*), respectively. Furthermore, we use a Gaussian dataset (*normal*) and a dataset built from two Gaussians with a sparse middle area (*dnormal*). Figure 4 shows their target distributions. We train models with DenseLoss and different α values to gain insight into the practical effects of different degrees of DenseWeight.

4.1.1 Dataset creation

We use an MLP as a random function to generate synthetic datasets. This guarantees that the function can be learned again by an MLP in theory. Our network's parameters are initialized with a standard Gaussian distribution. This network is provided with 200,000 sets of 10 features each. The features are also drawn from a standard Gaussian distribution. The network consists of 3 hidden layers with 10 neurons each and ReLU (Nair and Hinton 2010) activation. The final hidden layer is connected to a single neuron with linear activation to obtain target values for a regression task. From the resulting 200,000 data points 10,000 were sampled in such a way that there are uniformly distributed target values. This uniform dataset's target values range from − 32.13 to 76.42. Then, for each dataset a probability density function is defined corresponding to the desired target distribution. 1000

data points are sampled from the uniform dataset weighted by the samples' desired densities, creating the datasets *pareto*, *rpareto*, *normal*, and *dnormal*. Figure 4 visualizes their target variable distributions.

Each dataset is split randomly in a training (60%), validation (20%), and test (20%) set. The resulting splits are inspected to confirm that their target variables are similarly distributed. Otherwise it would be possible that sparsely sampled ranges in the target variable are not represented in a split through unfortunate random sampling.

4.1.2 Experimental setup

To illustrate how DenseLoss affects model performance for underrepresented parts of datasets based on our weighting scheme DenseWeight, we conduct a parameter study to examine the effects of different α values. Therefore, we train models with α values ranging from 0.0 to 2.0 with steps of 0.1. To strengthen confidence in the results of this experiment we train 20 model instances per α which are used for testing statistical significance with the Wilcoxon signed-rank test and a significance level of 0.05.

The MLP used is structurally equal to the data generator network. Thus, this model also consists of 3 hidden layers with 10 neurons each and ReLU activation as well as one neuron with linear activation for the output layer. Instead of initializing parameters from a standard Gaussian distribution, we use Kaiming Uniform initialization (He et al. 2015). DenseLoss is the loss function used in conjunction with the metric mean squared error (MSE). The model is trained with Adam optimization (Kingma and Ba 2014), a learning rate of 10^{-4}, and a weight decay coefficient of 10^{-9}. Training is run for at most 1000 epochs, but it is stopped early if the validation loss is not improving for 10 epochs in a row. This improves generalization performance (Prechelt 1998).

4.1.3 Results

To evaluate model performance for separate parts of the target domain, we bin the test data points based on their target value. Each bin spans 20% of the target variable's range in the test set. We rank these bins per dataset by the number of data points. The bin with the fewest (most) samples has bin rank 1 (5) and is called the least (most) common bin. This allows performance comparisons between similarly rare bins over all datasets. We calculate the root mean squared error (RMSE) and mean absolute error (MAE) for each individual model instance of the 20 instances per tested configuration.

Our MLP without DenseLoss achieves on average over the 20 runs RMSEs (MAEs) between 3.53 (2.70) and 6.75 (5.47) for the most common bins, i.e. bin rank 5, and between 6.68 (6.26) and 27.10 (26.74) for the rarest bins, i.e. bin rank 1, across the synthetic datasets. We find that DenseLoss with, for example, $\alpha = 1.0$ improves average RMSE (MAE) for the rarest bins by between 1.21 (1.48) and 7.02 (7.00) while increasing it for the most common bins by between 1.12 (0.90) and 1.68 (1.49).

Figure 5 visualizes the mean RMSE of models trained with different α values over all synthetic datasets for different bin ranks. DenseLoss typically improves performance in sparsely sampled bins (bin ranks 1–3) with a suitable α value. As expected, DenseLoss tends to reduce performance for bins with many samples (bin ranks 4 and 5). We find that most α values greater than 0 lead to improvements in rare bins. For example, for *pareto* all tested configurations with $\alpha \geq 0.8$ yielded improvements in the rarest bin and the same is true for all runs with $\alpha \geq 0.2$ for *dnormal*. For *rpareto* all runs with DenseLoss enabled

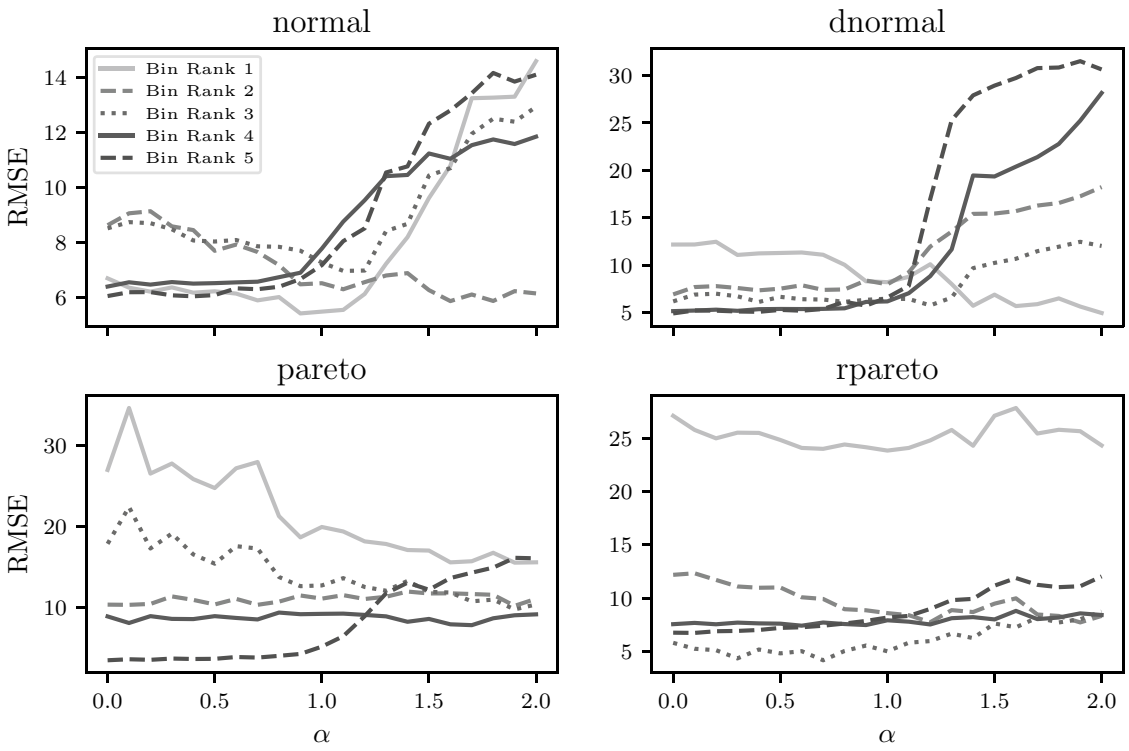


Fig. 5 Mean RMSE per α and bin rank over the synthetic datasets. Bins are ranked in each test set by sample size. Bins with rank 1 (5) contain the fewest (most) samples

($\alpha > 0.0$) improved the rarest bin except for $\alpha = 1.5$ and $\alpha = 1.6$, where performance dropped slightly. *normal*'s rarest bin is improved with $0.1 \leq \alpha \leq 1.2$, which is discussed in the next paragraph. As described at the beginning of Sect. 4 we conduct statistical significance tests to strengthen confidence in our results. When considering $\alpha = 1.0$, which seems to provide good performance for rare data points across all datasets, we find that the performance for the rarest bin has improved significantly compared to not using DenseLoss ($\alpha = 0.0$) for each dataset. Bin rank 2 is improved significantly with $\alpha = 1.0$ for *normal* and *rpareto*, while bin rank 3 is significantly better for *normal* and *pareto*. We also see with $\alpha > 1.0$ that the performance for the most common bin deteriorates considerably for *normal* and *dnormal*, as the weight of more and more of these data points is pushed towards ϵ. This effect is also noticeable in the other bin ranks albeit with reduced strength the rarer the bins get. Interestingly, this performance degradation seems less pronounced for both pareto datasets. We find very similar results with regards to the metric MAE.

Figure 6 shows detailed results for datasets *normal* and *pareto*. Bins are identified by bin rank and ordered to correspond to the dataset's distribution plot at the top, thus visualizing RMSE and density from the lowest (left-most bin) to the highest target values (right-most bin). Setting α to around 1 provides improved performance for rare target ranges while only slightly reducing performance for common target ranges. For example, with $\alpha = 1.0$ in *pareto* we observe an increase in RMSE of 1.68 in the most common bin with rank 5 and a drop in RMSE of 7.02 in the least common bin with rank 1. In general, error for samples in rare target ranges tends to decrease with increasing α while performance in common target ranges mostly deteriorates. For *normal*'s rarest bin with rank 1 too large α values ($\alpha \geq 1.4$) also show performance degradation. We hypothesize that this can occur when the target range in the training set has very few data points and the neighboring, more common data points are assigned weights close to 0. In this case the model seems to struggle to learn a

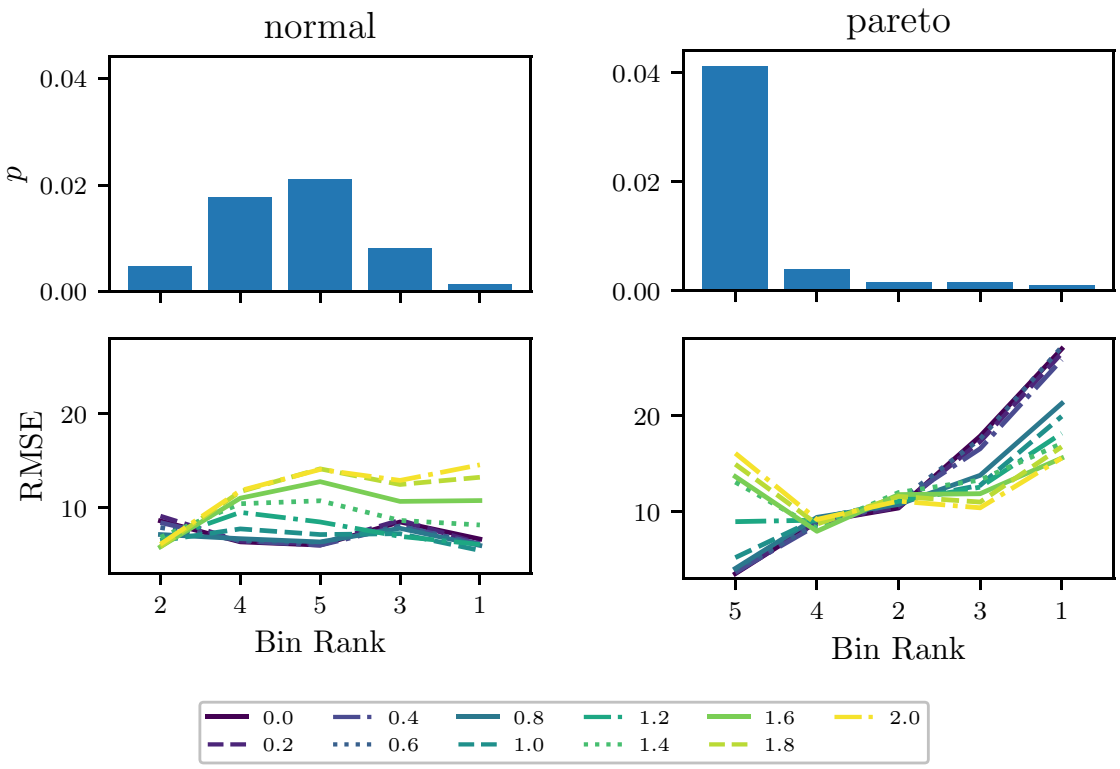


Fig. 6 Mean RMSE per test bin over 20 runs for datasets *normal* (left) and *pareto* (right). Bar charts show the density per bin in the test set. Line plots visualize the mean RMSE per test bin for the α values shown in the box at the figure's bottom

general function for the higher target ranges, because of the effectively small number of samples there.

Given the continuous nature of regression datasets it is also interesting to regard the performance over the datasets' target domains. To account for different distributions among the datasets for this evaluation we calculate the normalized density per test data point's target value (as defined in Eq. 2) through KDE (same parameters as for DenseWeight's KDE) on the target variable of its respective test dataset. In contrast to before, we do not use this normalized density to weight samples or train models but instead use it as a dataset-independent metric for each sample's rarity within its test dataset. This rarity thus provides us with a dataset-independent proxy of the target variable domains. It is independent from the rarity used during model training and does not influence the estimates for the test samples. Also, we calculate the MAE over the 20 runs for each test data point of each dataset. To enable a continuous evaluation over all datasets we normalize the MAE via division by the difference between the maximum and the minimum value of its respective test dataset's target variable. The normalized MAE in conjunction with the normalized densities allow us to plot Fig. 7 which visualizes the normalized MAE depending on the data point rarity across all datasets for regular training ($\alpha = 0.0$), DenseLoss ($\alpha = 1.0$ and $\alpha = 2.0$), and also the state-of-the-art imbalanced regression method SMOGN. To account for the high variability and to improve interpretability we smoothed the plot by applying a moving mean with a windows size of 30 data points over the 800 total test data points. We find that DenseLoss with both $\alpha = 1.0$ and $\alpha = 2.0$ typically reduces error for very rare samples ($\sim p'(y) < 0.15$). Performance with $\alpha = 2.0$ deteriorates considerably for more common data points ($\sim p'(y) > 0.4$) while performance of $\alpha = 1.0$ remains close to $\alpha = 0.0$ up until around $p'(y) > 0.75$ where a gap emerges.

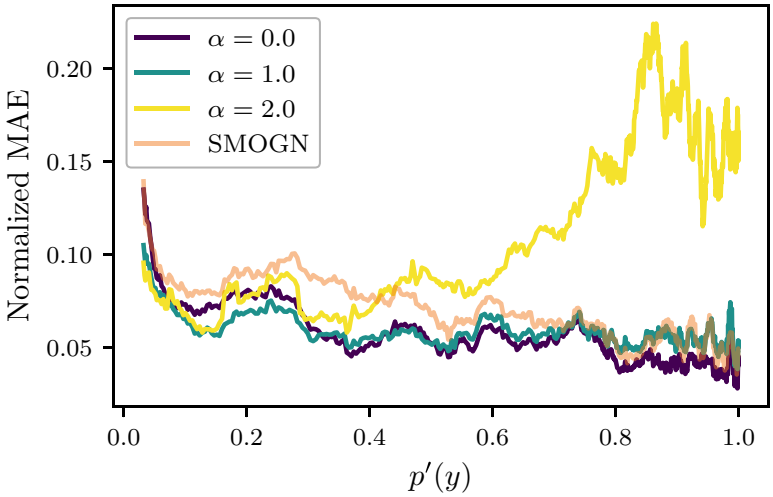


Fig. 7 Normalized MAE for test samples from all synthetic datasets per normalized density. Graph is smoothed via moving mean (window size 30) to ease interpretability

While this experiment mainly analyzes DenseLoss in a controlled manner we also applied SMOGN to our synthetic datasets, finding mostly better performance for $\alpha = 1.0$ than SMOGN, when applying SMOGN as described in Sect. 4.2. Rare parts in *pareto* and *rpareto* were identified automatically; rare parts in *normal* and *dnormal* were identified manually, since the automatic method wrongly deemed all samples relevant. For *normal* and *dnormal* we used the control points $(-10, 1, 0), (20, 0, 0), (50, 1, 0)$ and (0, 0, 0), (20, 1, 0), (50, 0, 0), respectively. Resulting relevance functions are visualized in the "Appendix". Since SMOGN's automatic method for obtaining ϕ only works for datasets where rare values are also extreme, it is not suited for *dnormal*. With our manual control points it is still not ideal as it incorrectly deems low target values as relevant, but it is substantially better than considering all data points relevant. *normal*'s manual ϕ shows no such issues. When considering binned evaluation we find that DenseLoss with $\alpha = 1.0$ performs significantly better than SMOGN for the rarest bin on all datasets except *pareto*.

This experiment confirms that DenseLoss allows shifting a model's focus to rarer cases away from the cases it would have focused on with regular training. Inspecting the model performance across the target range with varying α values enables an informed choice for the trade-off between performance in common and rare cases. Thus, DenseLoss provides additional control over model training, allowing to fit models with better performance for rare data points.

4.2 Comparison with state-of-the-art

SMOGN can currently be considered the state-of-the-art method for imbalanced regression, as it has shown to be better than the other available method SmoteR (Branco et al. 2017). SMOGN's authors present 20 imbalanced datasets in their paper. We apply both SMOGN and DenseLoss to those datasets and compare model performances. Neural networks trained without applying any method for imbalanced data are used as a baseline. To this end we apply both methods and the baseline to the 20 imbalanced datasets from SMOGN's test section (Branco et al. 2017). We obtain the data from their repository[2].

[2] https://github.com/paobranco/SMOGN-LIDTA17

See the "Appendix" for an overview. We also compared DenseLoss with SMOGN using DenseWeight for its relevance function in the "Appendix", finding similar results as presented in the following, where we compare DenseLoss to SMOGN using its default relevance function.

4.2.1 Experimental setup

As with the synthetic data, we randomly split each dataset in a training (60%), a validation (20%), and a test (20%) set. Considering the small size of some of the datasets, we inspect the splits to confirm that they are similarly distributed and redo the random split if the distributions are too different.

Models trained with DenseLoss use $\alpha = 1.0$. For SMOGN we use the python package *smogn* (Kunz 2019). Since SMOGN's authors also aim to increase performance for rare data points on these datasets we apply the same hyperparameters as they did in their paper: Rare target values are determined by their automatic method (Ribeiro 2011) as described in Sect. 2. Just as SMOGN's authors, we consider target values rare where the relevance function yields more than 0.8. SMOGN oversamples data points with rare target values to obtain a more balanced distribution. For oversampling SMOGN is set to consider the 5 nearest neighbor samples. The amount of Gaussian noise added for oversampling (i.e. perturbation) is set to 0.01. We use the same MLP architecture and hyperparameters as described in Sect. 4.1. Additionally, we repeat the experiment with the same hyperparameters but different MLP topologies, namely a deeper model (4 hidden layers with 10 neurons each), a shallower model (2 hidden layers with 10 neurons each), a wider model (3 hidden layers with 20 neurons each), and a narrower model (3 hidden layers with 5 neurons each), to confirm that our results are not due to a specific network architecture. We find very similar results for all architectures and therefore only report detailed results for one topology (3 hidden layers with 10 neurons each) for brevity. Models are trained and evaluated 20 times per dataset and method to test statistical significance with the Wilcoxon signed-rank test and a significance level of 0.05.

4.2.2 Results

As in Sect. 4.1, we split each test dataset into 5 equidistant bins and rank the bins by the number of samples. Metrics RMSE and MAE are calculated for each bin.

Figure 8 visualizes the number of dataset wins of DenseLoss, SMOGN, and the baseline (None) per bin rank over the 20 datasets for the metric RMSE. Due to some datasets' small sizes there are some bins without data points in the test set. This results in the bars for rank 1 and 2 not containing 20 wins, since no winner can be found for empty bins. The results show for the rarest bins (bin rank 1) that DenseLoss provides the best performance for 8 datasets while SMOGN only performs best on 3 datasets and applying no method is best for only 2 datasets. DenseLoss has the highest number of significant dataset wins against both methods in this rarest bin rank but also in bin ranks 2–4. For bin ranks 1–4, DenseLoss wins more than half of the datasets, with most wins being statistically significant against the baseline and SMOGN. Only for bin rank 5 with the most samples, it is typically best to apply no method for imbalanced data. This, however, is expected, as the usual training method is biased towards common target values. We found very similar results for the metric MAE. Repeating this experiment with the other network architectures introduced in Sect. 4.2.1 further confirms these findings, as is shown in the "Appendix". These

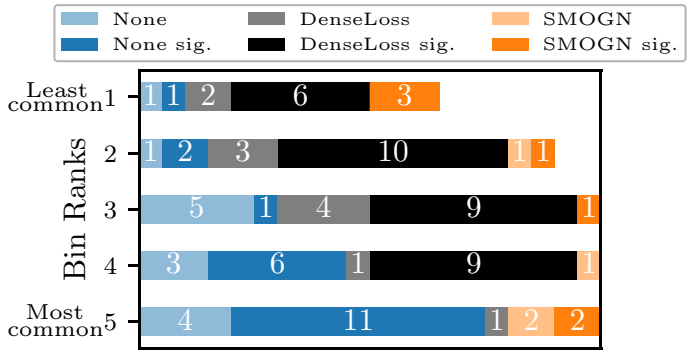


Fig. 8 Number of datasets won per method for each bin based on RMSE. Bins are ranked within each test dataset according to the number of data points. Bins with rank 1 (5) contain the fewest (most) samples. Each bar section shows the number of datasets won by a method at that bin rank. When a method's wins are denoted as "sig." they are significant with regards to both other methods. 5 test datasets had a bin without data points and 2 test datasets had 2 bins without samples. Because of this the bars for bin rank 1 and 2 are smaller as no winner can be determined for empty bins

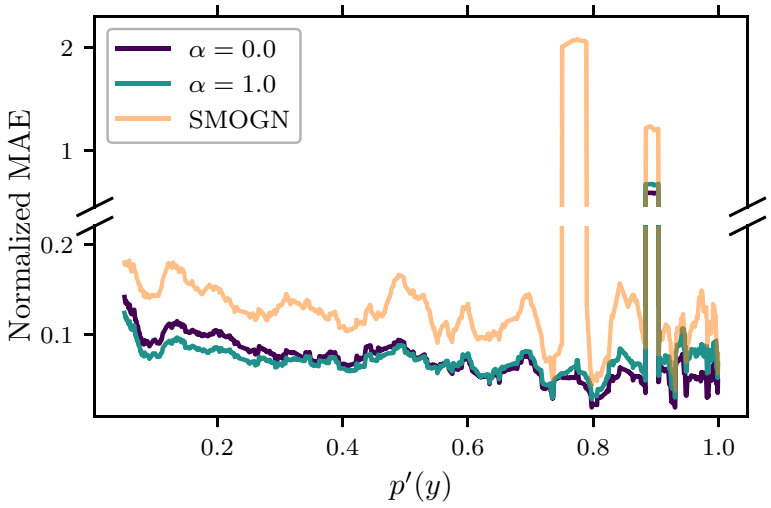


Fig. 9 Normalized MAE for test samples from all 20 datasets per normalized density. Graph is smoothed via moving mean (window size 300) to ease interpretability

results suggest that DenseLoss typically provides better performance for rare data points in comparison to the state-of-the-art imbalanced regression method SMOGN.

Similarly as described in Sect. 4.1.3 we analyze the performance over the datasets' target variable domains in a continuous manner. Thus, we visualize the normalized MAE per data point rarity across all datasets for regular training ($\alpha = 0.0$), DenseLoss ($\alpha = 1.0$), and SMOGN in Fig. 9. To account for the high variability and to improve interpretability we smooth the plot using a moving mean with a windows size of 300 samples over the 7188 total test samples. Similarly as in the bin-wise evaluation, we find on average lower error with DenseLoss for rarer data points ($\sim p'(y) < 0.5$) compared to using no imbalanced regression method. Normalized MAE is improved by roughly 10% for rare data points with $p'(y) < 0.3$ while the error increases with larger densities. SMOGN seems to not work well on average over all datasets even though we used the same datasets with the same hyperparameters as the original SMOGN authors used in their work. We find high variability in SMOGN's performance across the datasets with it working well for some datasets (e.g. cpuSm or acceleration) but considerably worse on most others, leading to relatively large normalized MAE regardless of density. Also note the two outlier segments in the plot showing high MAE that stem from one sample each of the dataset availPwr.

Almost all models estimate extremely large values for these two samples, likely due to an unusually high feature value, leading to very large MAE for all moving mean windows that include these samples.

4.3 Statistical downscaling of precipitation

To show that DenseLoss can work for larger datasets and more complex neural network architectures, we apply it to the real world task statistical downscaling of precipitation. Its objective is to generate local scale precipitation projections based on spatially coarse precipitation projections stemming from Earth System Models. This can be learned based on high-resolution historical climate observations (Vandal et al. 2017).

A model that does statistical downscaling of precipitation is DeepSD (Vandal et al. 2017). It uses super-resolution convolutional neural networks to improve the resolution of precipitation data. The model is supplied with a map showing daily precipitation at a low spatial resolution. This map is similar to an image where each pixel contains precipitation data for a specific real world area. Additionally, the model is provided with a high-resolution elevation map whose pixels are aligned with the precipitation map, so that any pixel in one map represents the same area as the corresponding pixel in the other map. This information helps the model to take topography as a known influence into account (Daly 2008). DeepSD's authors use the PRISM dataset (Daly 2008) for precipitation data over the Continental United States and elevation data from the GTOPO30 dataset (U.S. Geological Survey 1996). Commonly, there are far less rainy days than dry days at most locations (see "Appendix"). Yet, especially high precipitation events are interesting as they could have considerable consequences like flooding. Thus, we apply DenseLoss to this real world task in order to improve model performance especially for these rare and extreme events. To this end, we conduct a study for DenseLoss 's α, investigating the influence of α on model performance.

4.3.1 Experimental setup

For our study, we modified DeepSD's code[3] to include DenseLoss. As such, we use three convolutional layers with 64, 32, and 1 filters and kernel sizes of 9, 1, and 5, respectively. Model training minimizes the MSE with a batch size of 200 and the Adam optimizer with a learning rate of 10^{-4} for the first two layers and 10^{-5} for the last layer. Precipitation data is split into a training (years 1981 to 2005) and a test set (years 2006–2014). In contrast to the hyperparameter values described in the DeepSD paper, we trained for 10^5 instead of 10^7 epochs. This saves computation time as we found no further reduction in training loss when training longer. These are still many epochs but it is necessary given the relatively low learning rates used by DeepSD. We train DeepSD to downscale from 128 to 64 km resolution. The study tests α values from 0.0 to 4.0 with steps of 0.2. Compared to Sect. 4.1 we extend this range to assess at which α performance plateaus considering we have found continuous performance gains up to $\alpha = 2.0$ here. DeepSD is trained 20 times per α with different random model initializations to test statistical significance with the Wilcoxon signed-rank test and a significance level of 0.05 (Vandal et al. 2017).

[3] https://github.com/tjvandal/deepsd.

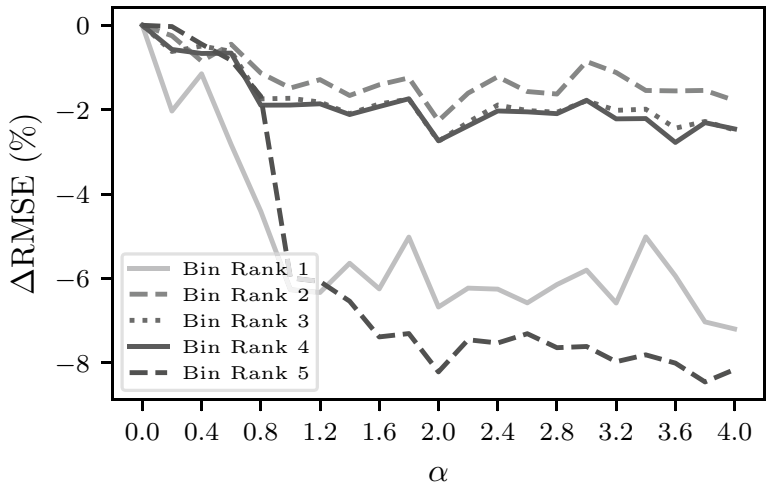


Fig. 10 Change in mean RMSE with respect to not using DenseLoss ($\alpha = 0.0$) per α for each bin rank in PRISM's test set. Bins are ranked within the test dataset according to the number of samples. The Bin with rank 1 (5) contain the fewest (most) samples

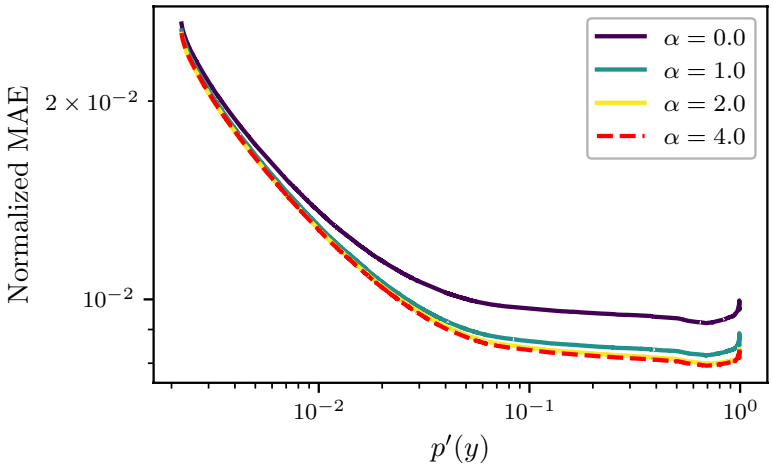


Fig. 11 Normalized MAE for PRISM test samples per normalized density. Graph is smoothed via moving mean (window size 300,000) and logarithmic for interpretability

4.3.2 Results

As before we split the test dataset into 5 equidistant bins, rank the bins by the number of samples and calculate RMSE and MAE for each bin.

Figure 10 visualizes the change of mean RMSE in percent with respect to regular training ($\alpha = 0.0$) in all bin ranks for models trained with different α. E.g. a ΔRMSE of -8% indicates an 8% lower RMSE compared to not using DenseLoss. Interestingly, DenseLoss does not only improve performance for rare samples (e.g. bin rank 1) but also for common values (e.g. bin rank 5) here. Improvement is most pronounced in the most common and the rarest bin. This suggests that the enormous over-representation of samples with precipitation close to 0 mm may also negatively affect performance for these same very common data points. DenseLoss reduces their influence, effectively reducing the over-representation which in turn seems to lead to better performance for common samples. Performance improves with increasing α before plateauing for $\sim \alpha \geq 2.0$. Our tests for statistical significance show that for each $\alpha \geq 0.8$ performance improvements compared to $\alpha = 0.0$ are significant for all bin ranks.

As in the previous experiments (e.g. Sect. 4.1.3) we also analyze the performance over the target variable domain in a continuous manner. Thus, we visualize the normalized MAE per sample rarity for regular training ($\alpha = 0.0$) and DenseLoss ($\alpha > 0.0$) in Fig. 11. To improve interpretability we smooth the plot by applying a moving mean with a window

size of 300,000 data points over the 6,143,403 test samples. We see again that DenseLoss improves estimates for both rare (left side) and even more so for common samples (right side) here. Performance improvements tend to increase with larger α but only marginally above $\alpha = 2.0$.

In this experiment, we observe a different behavior of DenseLoss than before. Here, DenseLoss is able to improve performance across the complete target variable range instead of trading performance between common and rare samples. We hypothesize that DeepSD's capacity is large enough to learn a good function for both rare and common data points at once, while smaller models might lack the capacity for this. DenseLoss seems to allow this model to converge to an overall better solution.

5 Discussion

In this work, we have shown that DenseWeight and DenseLoss can help to improve model performance for rare data points. However, there are still aspects to discuss.

While DenseWeight can theoretically be used with any algorithm that supports sample weights, we only evaluated it with neural networks using DenseLoss. We expect to see similar results for other algorithms but we did not test this assumption here.

We compared our approach to SMOGN in Sect. 4.2 but not in the last experiment as we found it to be computationally infeasible. SMOGN's oversampling algorithm calculates the distance between all data points where a data point is the precipitation at one location at one time. Using the available implementation we found through initial testing that this would take years with any hardware available to us.

We did not systematically check whether the architectures used in the first two experiments generalize well but we expect decent generalization performance due to our use of early stopping. Model training is stopped when the validation loss stops improving, which inhibits overfitting. Its effectiveness shows in spot checks where we found no model with substantially higher training than test or validation performance.

Our approach introduces a new hyperparameter α, controlling the strength of density-based weighting, which must be set appropriately. While we find that setting $\alpha = 1.0$ typically provides good performance for rare samples, there can be better choices. With a validation dataset and a suitable goal it is possible to optimize α, however defining a goal is often not trivial in an imbalanced regression setting. It requires domain knowledge to define which data points are rare and important. If this knowledge is available one could simply search for the α that minimizes the MSE on these rare and important data points to achieve optimal performance for a specific domain.

For the data splits in Sect. 4.2 we manually confirmed whether the splits are similarly distributed. Random splitting was not able to reliably produce splits with similar distributions given the small sizes of some datasets. While we do not believe this to influence the results, a more automatic method to this would be more objective. One could perhaps try to maximize a distribution similarity score and stop redoing splits if a certain threshold is exceeded but we did not implement this in this work.

6 Conclusion

In this work, we have proposed our sample weighting approach for imbalanced regression DenseWeight and our cost-sensitive learning method DenseLoss, tackling the problem of imbalanced regression for neural networks based on DenseWeight. We show that our approach can improve model performance for rare data points with synthetic datasets, specifically designed to represent different kinds of data distributions. Extensive hyperparameter studies for each dataset provide insight and intuition for how DenseWeight and DenseLoss 's α controls a model's focus on rare in comparison to common data points. Experiments on 20 datasets show that DenseLoss typically outperforms the sampling-based method SMOGN. Applying DenseLoss to statistical downscaling of precipitation, we demonstrate its benefits on a real world task and discuss its potential for higher capacity models.

Future work includes examining ensemble approaches for DenseLoss which combine models trained with different α. Depending on α, each model is an expert in different target variable ranges. A meta-model could learn which ensemble member is likely to perform best based on a given sample's features which may lead to nearly optimal performance not only for rare samples but across the whole target range. Furthermore, it is interesting to assess the relation between model capacity and performance across the target domain with DenseLoss, following the intuition that large enough models might be able to learn a function that consistently works well for both rare and common data points. Additionally, ideas which are already established for cost-sensitive learning in imbalanced classification settings could be transferred for regression tasks. Examples include weighting based on the effective number of samples in a target region (Cui et al. 2019) or incorporating sample difficulty in the weighting (Dong et al. 2017).

Appendix

In the "Appendix" we present some additional details and results of our work.

Figure 12 visualizes detailed results for datasets *dnormal* and *rpareto* of our experiment with synthetic data. Table 1 lists the datasets used in our comparison to the state-of-the-art with their respective sizes. Figure 13 visualizes the relevance functions for SMOGN on the synthetic datasets. Figure 14 shows the number of datasets won per method in our comparison with the state-of-the-art for additional network architectures. Figure 15 depicts the highly skewed precipitation distribution of the PRISM dataset used in our experiment for statistical downscaling of precipitation.

SMOGN with DenseWeight

The results in this work show that DenseLoss typically outperforms SMOGN. However, it is not clear to which extent the performance differences stem from the different measures of data point rarity or from the methodological differences between resampling and cost-sensitive learning. We therefore adapt SMOGN to use DenseWeight as its relevance function and we repeat the experiments involving SMOGN. We call SMOGN with DenseWeight *SMOGN-DW* in the following.

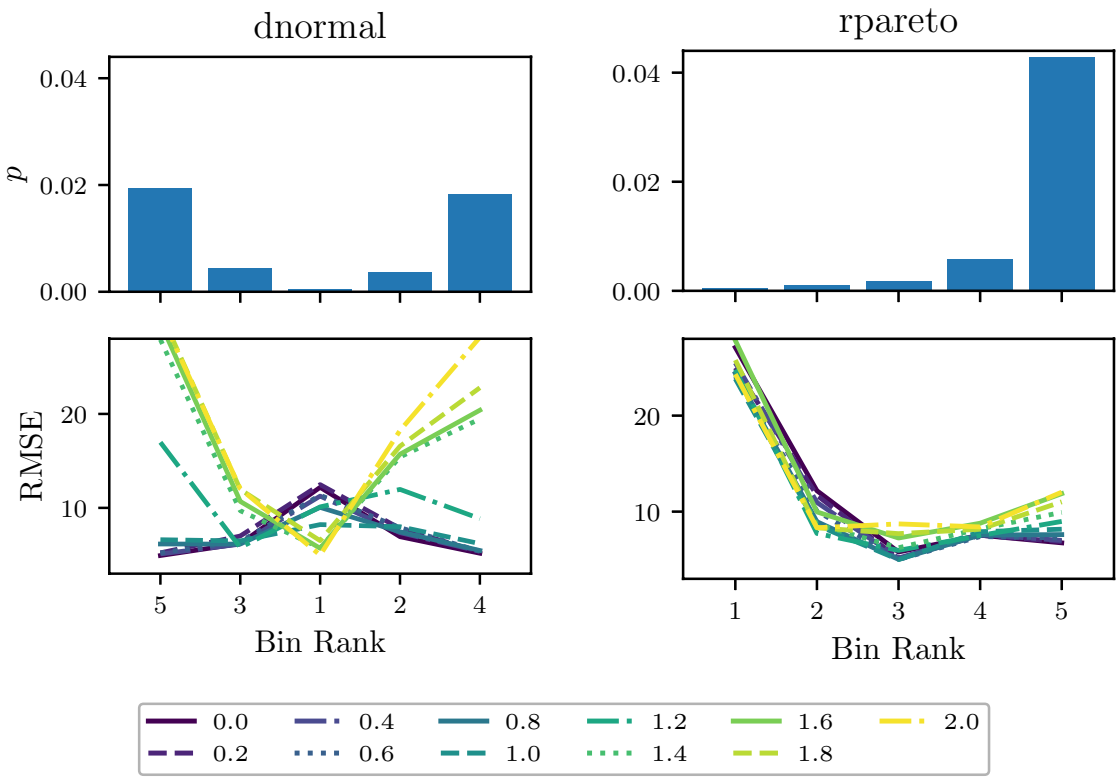


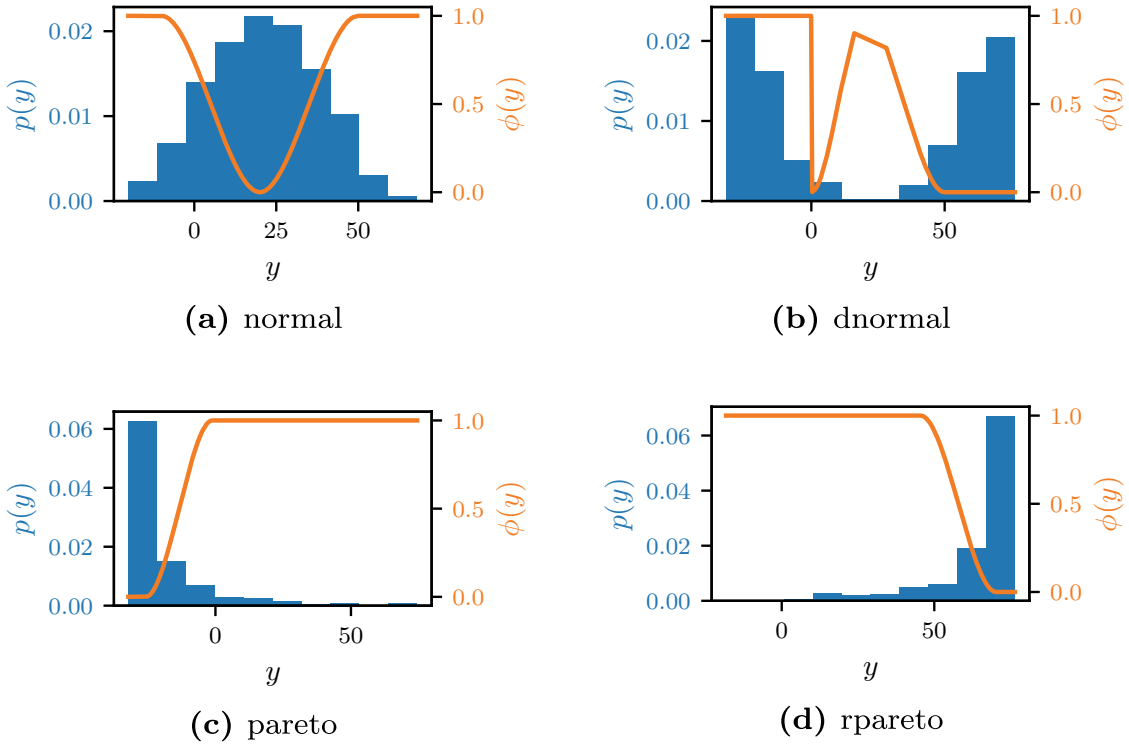
Fig. 12 Mean RMSE per test bin over 20 runs for datasets *dnormal* on the left and *rpareto* on the right. The top charts show the density per bin in the respective test dataset, visualizing the target variable's distribution. The line plots below visualize the mean RMSE per test bin for the α values shown in the box at the figure's bottom

Table 1 Datasets with imbalanced target values and their sizes

Dataset	*N*	Dataset	*N*	Dataset	*N*	Dataset	*N*
a1	198	a6	198	AvailPwr	1802	dAiler	7129
a2	198	a7	198	Bank8FM	4499	FuelCons	1764
a3	198	Abalone	4177	Boston	506	MachineCpu	209
a4	198	Acceleration	1732	ConcrStr	1030	MaxTorq	1802
a5	198	Airfoild	1503	cpuSm	8192	Servo	167

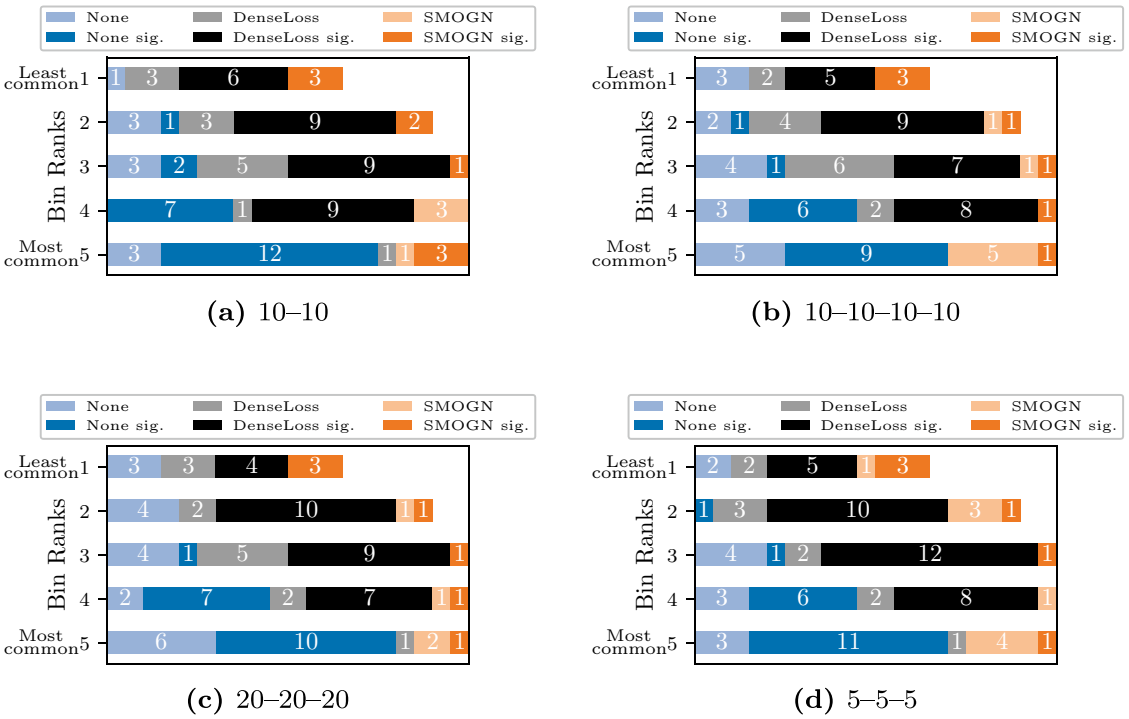
Experimental Setup

SMOGN identifies the rarity of each data point through a relevance function $\phi : Y \mapsto [0, 1]$ which is obtained through an automatic method based on box plot statistics by SMOGN's authors (Ribeiro 2011; Branco et al. 2017). This relevance function is similar to DenseWeight in that both aim to measure the rarity of a data point. In order to use DenseWeight as a relevance function we normalize the weights of all training set data points to a range between 0 and 1. We set DenseWeight's α to 1 which is the same value used for DenseLoss in the comparisons with SMOGN. We implement SMOGN with DenseWeight by expanding the existing python implementation *smogn* (Kunz 2019). This expanded *smogn* package is also available in our online repository. All other aspects of the experimental setup remain as described in Sect. 4.



(a) normal (b) dnormal (c) pareto (d) rpareto

Fig. 13 SMOGN's relevance function ϕ for the synthetic datasets



(a) 10–10 (b) 10–10–10–10 (c) 20–20–20 (d) 5–5–5

Fig. 14 Number of datasets won per method and bin based on RMSE with different MLP architectures. Subcaptions indicate the number of hidden layers and neurons per hidden layer. E.g., 10–10 represents an MLP with two hidden layers each having 10. Bins are ranked in each test dataset according to sample size. Bins with rank 1 (5) contain the fewest (most) samples. Bar section's show the number of datasets won by a method at that bin rank. Wins denoted as "sig." are significant with regards to both other methods. 5 test datasets had an empty bin and 2 test datasets had 2 empty bins. Thus, the bars for bin rank 1 and 2 are smaller as there is no winner for empty bins

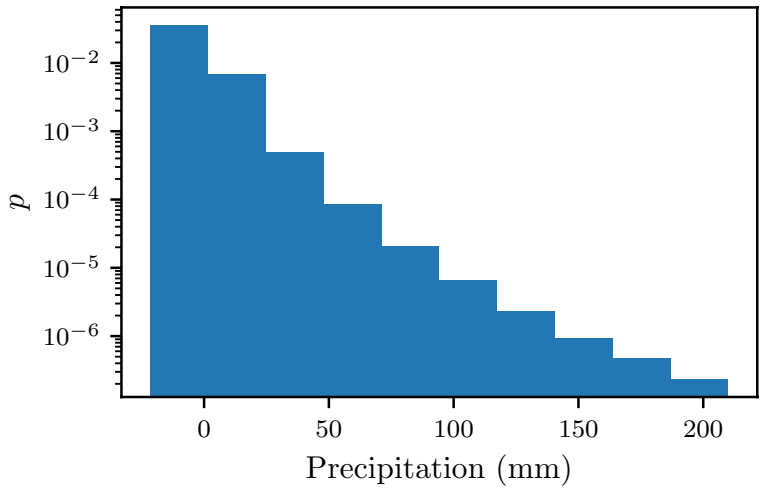


Fig. 15 Distribution of precipitation in the PRISM dataset over all cells and all days from 1981 to 2005. Note that the y-axis is logarithmic. Negative precipitation values may stem from an interpolation method used in the original work, but we decided not to clean the data to stay consistent with previous work

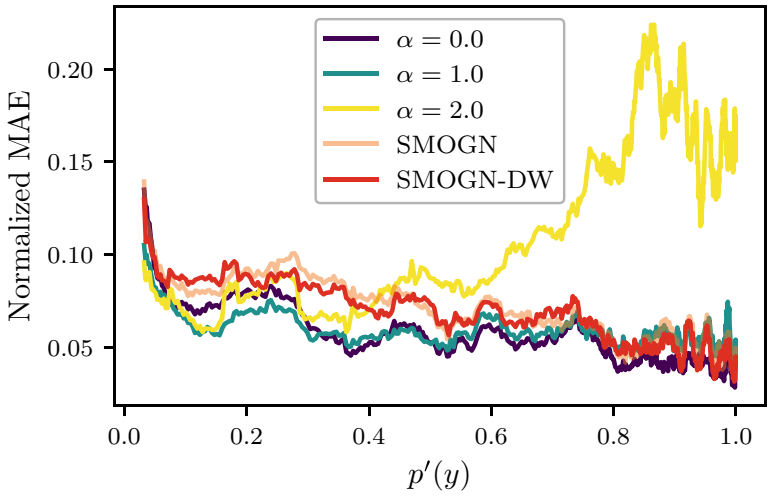


Fig. 16 Normalized MAE for test samples from all synthetic datasets per normalized density. Graph is smoothed via moving mean (window size 30) to ease interpretability

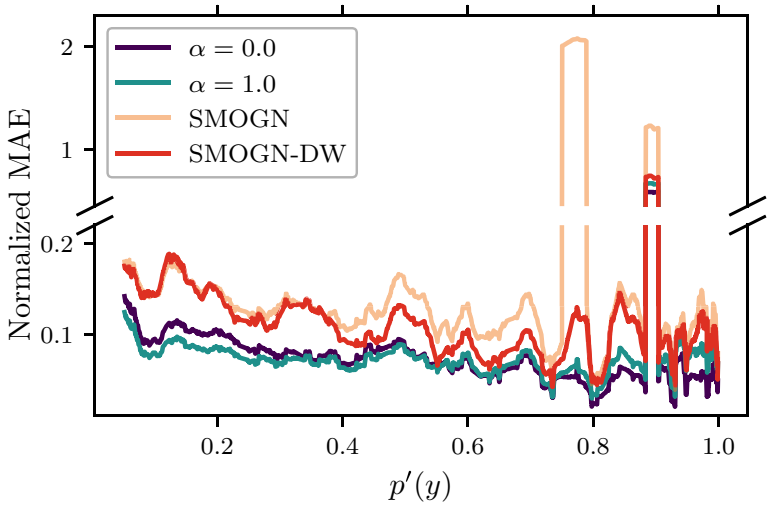


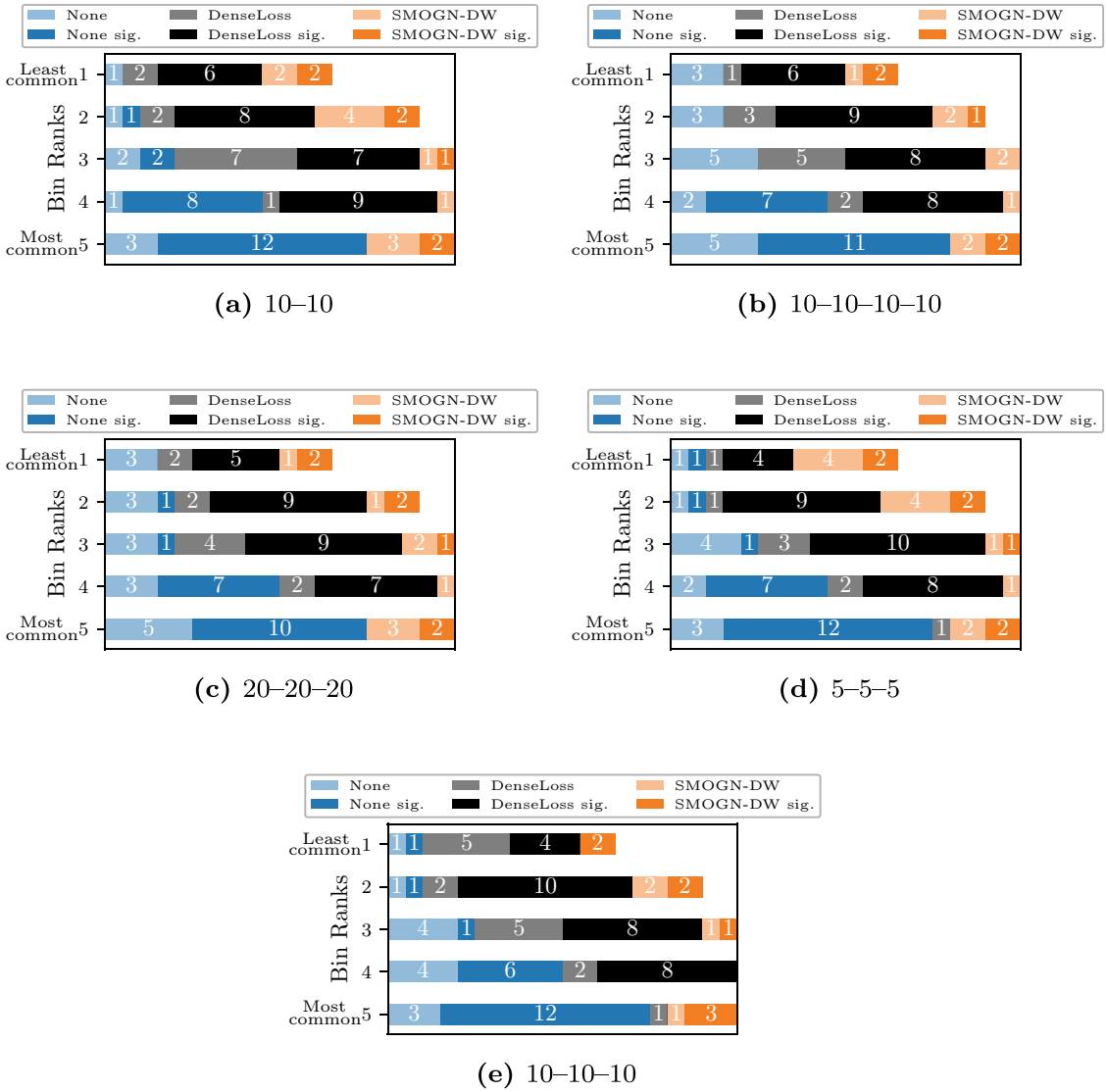
Fig. 17 Normalized MAE for test samples from all 20 datasets per normalized density. Graph is smoothed via moving mean (window size 300) to ease interpretability

Results

Figure 16 shows the normalized MAE depending on test data point rarity over the synthetic datasets (as in Fig. 7) now also with SMOGN-DW. We see that SMOGN and SMOGN-DW perform very similarly on these synthetic datasets and that DenseLoss still tends to provide

better performance for more rare data points. For the continuous results over the twenty datasets from Sect. 4.2 we see in Fig. 17 that SMOGN and SMOGN-DW also show mostly similar performance, with lower normalized MAE for data points with $\sim 0.7 > p'(y) > 0.4$ for the latter. DenseLoss still seems to provide better performance than SMOGN-DW and SMOGN.

Figure 18 shows the number of dataset wins (as in Fig. 8) but now with SMOGN-DW instead of regular SMOGN and for all evaluated MLP architectures. DenseLoss still has the highest number of significant dataset wins against both methods and almost always wins more than half of the datasets for bin ranks 1 to 4. Only bin rank 1 with architecture 5–5–5 shows one more SMOGN-DW win than the DenseLoss wins but even there DenseLoss has more significant wins. When comparing these results with the dataset wins of regular SMOGN in Figs. 8 and 14 we see that the performance difference between SMOGN and SMOGN-DW is rather small with SMOGN-DW occasionally competing slightly better.



(a) 10–10

(b) 10–10–10–10

(c) 20–20–20

(d) 5–5–5

(e) 10–10–10

Fig. 18 Number of datasets won per method and bin based on RMSE with different MLP architectures. Subcaptions indicate the number of hidden layers and neurons per hidden layer. E.g., 10–10 represents an MLP with two hidden layers each having 10. Bins are ranked in each test dataset according to sample size. Bins with rank 1 (5) contain the fewest (most) samples. Bar section's show the number of datasets won by a method at that bin rank. Wins denoted as "sig." are significant with regards to both other methods. 5 test datasets had an empty bin and 2 test datasets had 2 empty bins. Thus, the bars for bin rank 1 and 2 are smaller as there is no winner for empty bins

Since using the same measure of rarity for both SMOGN and DenseLoss does not improve SMOGN's performance considerably, we can conclude that most of the performance difference seems to stem from the methodological differences between resampling and cost-sensitive learning. Using DenseWeight as a relevance function for SMOGN seems to provide slight improvements compared to the relevance function used by SMOGN's authors but not enough to close the gap to DenseLoss.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Branco, P., Ribeiro, R. P., & Torgo, L. (2016a). UBL: An R package for utility-based learning. arXiv preprint arXiv:1604.08079.

Branco, P., Torgo, L., & Ribeiro, R. P. (2017). SMOGN: A pre-processing approach for imbalanced regression. In *LIDTA*.

Branco, P., Torgo, L., & Ribeiro, R. P. (2016b). A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR), 49*(2), 1–50.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority oversampling technique. *JAIR, 16,* 321–357.

Chen, Y.-C. (2017). A tutorial on kernel density estimation and recent advances. *Biostatistics and Epidemiology, 1*(1), 161–187.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., & Belongie, S. (2019). Class-balanced loss based on effective number of samples. *CVPR, 2018,* 9268–9277.

Daly, C., et al. (2008). Physiographically sensitive mapping of climatological temperature and precipitation across the conterminous United States. *International Journal of Climatology, 28*(15), 2031–2064.

Dong, Q., Gong, S., & Zhu, X. (2017). Class rectification hard mining for imbalanced deep learning. *ICCV, 2017,* 1851–1860.

Grinstead, C. M., & Snell, J. L. (2012). Introduction to probability. AMS.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *IJCNN 2008*. IEEE (pp. 1322–1328).

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV* 2015.

Hernández-Orallo, J. (2014). Probabilistic reframing for cost-sensitive regression. In *TKDD 8.4*.

Hernández-Orallo, J. (2013). ROC curves for regression. *Pattern Recognition, 46*(12), 3395–3411.

Huang, C., Li, Y., Change Loy, C., & Tang, X. (2016). Learning deep representation for imbalanced classification. *CVPR, 2016,* 5375–5384.

Kamalov, F. (2020). Kernel density estimation based sampling for imbalanced class distribution. *Information Sciences, 512,* 1192–1201.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence, 5*(4), 221–232.

Kunz, N. (2019). Smogn. [Online; version 0.1.2]. https://git.io/JOWoK.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. *ICML, 2010,* 807–814.

Odland, T. (2019). KDEpy. [Online; version 1.0.10]. https://git.io/JOWrM.

Prechelt, L. (1998). Early stopping-but when? In *Neural networks: Tricks of the trade* (pp. 55–69). Springer.
Ribeiro, R. P. (2011). Utility-based Regression. PhD thesis. University of Porto.
Ribeiro, R. P., & Moniz, N. (2020). Imbalanced regression and extreme value prediction. *Machine Learning, 109*(9), 1803–1835.
Silverman, B. W. (1986). *Density estimation for statistics and data analysis* (Vol. 26). CRC Press, London
Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *IJPRAI, 23*(04), 687–719.
Torgo, L., Ribeiro, R. P., Pfahringer, B., & Branco, P. (2013). Smote for regression. In *Portuguese conference on artificial intelligence* (pp. 378–389). Springer.
U.S. Geological Survey. (1996). GTOPO30. https://doi.org/10.5066/F7DF6PQS.
Vandal, T., Kodra, E., Ganguly, S., Michaelis, A., Nemani, R., & Ganguly, A. R. (2017). Deepsd: Generating high resolution climate change projections through single image super-resolution. *KDD, 2017,* 1663–1672.
Wang, Y.-X., Ramanan, D., & Hebert, M. (2017). Learning to model the tail. *NIPS, 2017,* 7029–7039.
Wilcoxon, F. (1945). Individual comparisons by ranking methods. In *Biometrics bulletin 1.6* (pp. 80–83). http://www.jstor.org/stable/3001968.
Zhao, H., Sinha, A. P., & Bansal, G. (2011). An extended tuning method for cost sensitive regression and forecasting. In *Decision support systems 51.3*.

Authors and Affiliations

Michael Steininger[1] · Konstantin Kobs[1] · Padraig Davidson[1] · Anna Krause[1] · Andreas Hotho[1]

Konstantin Kobs
kobs@informatik.uni-wuerzburg.de

Padraig Davidson
davidson@informatik.uni-wuerzburg.de

Anna Krause
anna.krause@informatik.uni-wuerzburg.de

Andreas Hotho
hotho@informatik.uni-wuerzburg.de

[1] Chair of Computer Science X, University of Würzburg, Würzburg, Germany

B. Publications with Limited Contribution

Atmospheric Environment 233 (2020) 117535

Contents lists available at ScienceDirect

Atmospheric Environment

journal homepage: http://www.elsevier.com/locate/atmosenv

OpenLUR: Off-the-shelf air pollution modeling with open features and machine learning

Florian Lautenschlager [a,*], Martin Becker [b], Konstantin Kobs [a], Michael Steininger [a], Padraig Davidson [a], Anna Krause [a], Andreas Hotho [a]

[a] Chair for Data Science, Computer Science, University of Würzburg, Am Hubland, 97074 Würzburg, Germany
[b] Stanford University, USA

HIGHLIGHTS

- Introduction of globally and openly available features for land use regression (LUR).
- Machine learning featuring automated hyper-parameter tuning for LUR tasks.
- Global features significantly enhance LUR through cross-learning on multiple cities.
- Source code and data available at dmir.org/openlur

ARTICLE INFO

Keywords:
LUR
Land use regression
Pollution
OpenStreetMap
Automated machine learning

ABSTRACT

To assess the exposure of citizens to pollutants like NO_x or particulate matter in urban areas, land use regression (LUR) models are a well established method. LUR models leverage information about environmental and anthropogenic factors such as cars, heating, or industry to predict air pollution in areas where no measurements have been made. However, existing approaches are often not globally applicable and require tedious hyper-parameter tuning to enable high quality predictions. In this work, we tackle these issues by introducing *OpenLUR*, an off-the-shelf approach for modeling air pollution that (i) works on a set of novel features solely extracted from the globally and openly available data source OpenStreetMap and (ii) is based on state-of-the-art machine learning featuring automated hyper-parameter tuning in order to minimize manual effort. We show that our proposed features are able to outperform their counterparts from local and closed sources, and illustrate how automated hyper parameter tuning can yield competitve results while alleviating the need for expert knowledge in machine learning and manual effort. Importantly, we further demonstrate the potential of the global availability of our features by applying cross-learning across different cities in order to reduce the need for a large amount of training samples. Overall, OpenLUR represents an off-the-shelf approach that facilitates easily reproducible experiments and the development of globally applicable models.

1. Introduction

Epidemiological studies show the negative impact of air pollutants like NO_x or particulate matter (UFP, $PM_{2.5}$ and PM_{10}) on respiratory and cardiovascular health (Pope et al., 1991; Polichetti et al., 2009; Brook et al., 2010). In order to assess the exposure of citizens to such pollutants, many measurement campaigns have been conducted. However, such campaigns are often restricted to very few stationary monitoring sites (Briggs et al., 2000; Carr et al., 2002; Brauer et al., 2003; Sahsuvaroglu et al., 2006; Henderson et al., 2007; Arain et al., 2007; Aguilera et al., 2007; Su et al., 2009; Dons et al., 2013; Ragettli et al., 2014; Montagne et al., 2015; Muttoo et al., 2018; Araki et al., 2018), and even if mobile monitoring devices are used, spatial coverage is limited to road segments or locations that have been chosen for the measurement campaign (Sîrbu et al., 2015; Larson et al., 2009; Zwack et al., 2011; Patton et al., 2014; Hasenfratz et al., 2014; Hankey and Marshall, 2015; Su et al., 2015; Shi et al., 2016; Minet et al., 2017; Basu et al., 2019). To retrieve pollutant concentration in unmeasured locations researchers rely on the correlation of air pollution with environmental and anthropogenic factors such as cars, streets, heating or industry (Jerrett et al.,

* Corresponding author. LSX, Computer Science, University of Würzburg, Am Hubland, 97074, Würzburg, Germany.
E-mail address: lautenschlager@informatik.uni-wuerzburg.de (F. Lautenschlager).

https://doi.org/10.1016/j.atmosenv.2020.117535
Received 15 November 2019; Received in revised form 15 April 2020; Accepted 17 April 2020
Available online 7 May 2020
1352-2310/

2004). In particular, they employ land use regression (LUR) models which leverage features extracted from land use statistics to overcome the limits and predict air quality in a spatially dense manner.

1.1. Problem setting

In previous work, the corresponding features usually stem from very specialized sources like local governments (Brauer et al., 2003; Hoek et al., 2001; Stafoggia et al., 2019), commercial providers (Sahsuvaroglu et al., 2006; Muttoo et al., 2018; Stafoggia et al., 2019), other models (for example traffic or weather models) (Dons et al., 2013; Stafoggia et al., 2019), or custom recordings (Briggs et al., 2000; Carr et al., 2002). For some studies, the source of the underlying land use data is even not easy to access (Montagne et al., 2015; Hankey and Marshall, 2015; Araki et al., 2018). The proposed methods are consequently hard to reproduce and hardly generalize to arbitrary locations.

Additionally, current work is often based on relatively simple models like linear regression (Arain et al., 2007; Aguilera et al., 2007; Muttoo et al., 2018) or generalized additive models (GAM) (Hasenfratz et al., 2014). While some newer work explores more advanced methods (Champendal et al., 2014; Brokamp et al., 2017; Araki et al., 2018; Stafoggia et al., 2019; Basu et al., 2019), state-of-the-art machine learning approaches are still frequently neglected or require tedious hyper-parameter studies.

1.2. Approach

In this work, we address this issue and propose *OpenLUR*, an off-the-shelf solution for air pollution modeling using land use regression (LUR) based on open features and state-of-the-art machine learning (see Fig. 1). First, to ensure reproducible and generalizable models, we derive features solely from openly and globally available data extracted from OpenStreetMap (OSM) (OpenStreetMap contributors, 2017). Second, we apply various state-of-the-art machine learning methods on these features. Besides GAMs and random forests, we specifically focus on methods that feature automated hyper-parameter tuning, for example AutoML (Blum et al., 2015), in order to eliminate the need for tiresome hyper-parameter studies. We evaluate both, our novel features as well as the state-of-the-art methods, on two large scale datasets: mobile air pollution data collected by Hasenfratz et al. (2014) and modelled air pollution data from the London atmospheric emissions inventory (Greater London Authority, 2016). We are able to show (i) that our novel open features outperform previously applied local feature sets on the given data, (ii) that using machine learning with automated hyper-parameter tuning yields high quality, reproducible and spatially generalizable models, (iii) that our features are applicable wherever OpenStreetMap data is available and (iv) that cross-learning on multiple cities can significantly enhance the model performance for small datasets.

1.3. Contribution

Our contributions in this article are: (i) We introduce a set of globally and openly available features for modeling air pollution using land use regression that significantly outperform previously proposed specialized features and show their global applicability. (ii) We evaluate state-of-art machine learning featuring automated hyper-parameter tuning for the application in land use regression tasks. (iii) We assess the enhancements for urban land use regression models achieved by the utilization of data from multiple cities. (iv) We propose OpenLUR as a globally applicable and expendable approach for land use regression and make the source code and our extracted features openly available at https://www.dmir.org/OpenLUR in order to ensure reproducibility and to enable future research.

2. Air quality training data

Our approach is generally applicable to any land use regression scenario. In this work, we train and test our models and features on a year of data collected during the OpenSense project in Zurich starting from April of 2012. Ensuing we show the global applicability of the approach on data extracted from the London Atmospheric Emissions Inventory (LAEI) (Greater London Authority, 2016) and demonstrate the potential of globally available land use features by combining both datasets.

2.1. OpenSense data

In the OpenSense project UFP was continuously measured by sensorboxes fixed to the top of tram cars (Hasenfratz et al., 2014). Hasenfratz et al. (2014) show the good measurement quality through the statistical distribution of measurements, comparison of baseline signals from several measurement devices and evaluation against high-quality datasets. With regard to preprocessing, we follow Hasenfratz et al. (2014): To rule out effects of seasonal variability on air pollution we split the collected data into four seasons of three months each (see Table 1).

To further smooth over smaller temporal and spatial variabilities and outliers, we divided the observation area into squares of $100m \times 100m$ and averaged the measurements for each season and square. Finally, squares with small numbers of samples which are prone to outliers and may negatively impact the model building process were removed. In particular, we kept the 200 squares with the largest amount of measured points (Hasenfratz et al., 2014). The # rows in Table 1 show the mean, min and max amount of measurements in the squares, that were kept in the dataset. The values used for the model training therefore are averages of at least 2000 single measurements which limits the influence of single outliers in the original data.

Table 1 shows statistics of our dataset by season. The mean as well as the standard deviation (SD) tend to be higher for the two later seasons in this dataset.

For a spatial visualization of Season 2, see Fig. 2. The particular

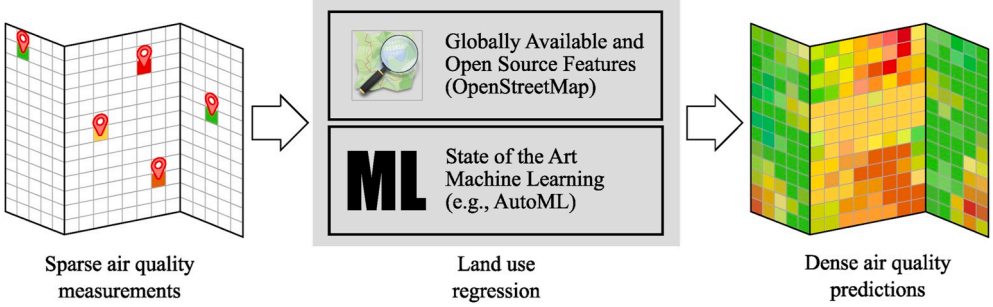


Fig. 1. Abstract-/ToC-Art: Off-the-shelf approach to air pollution modeling using land use regression (LUR) powered by openly available features and state-of-the-art machine learning: On the left this figure shows a set of sparsely collected air quality measurements. To derive a spatially dense map, we train a LUR air quality model using globally and openly available features derived from OpenStreetMap by applying state-of-the-art machine learning featuring automated hyper-parameter tuning. OpenLUR ensures easily reproducible experiments and enables world wide applicable models.

Table 1
The four seasonal OpenSense UFP datasets from Zurich and basic statistics. The # rows show mean, min and max count of measurements used for the average in the squares in which we aggregated the air pollution mesurements.

Season	1	2	3	4
From	April 01, 2012	July 01, 2012	October 01, 2012	January 01, 2013
To	June 30, 2012	September 30, 2012	December 31, 2012	March 31, 2013
Mean $[\frac{10^9 particles}{m^3}]$	12.88	13.69	16.08	17.99
SD $[\frac{10^9 particles}{m^3}]$	2.81	2.36	3.72	4.25
Mean #	7292	6111	11712	10986
Min #	2817	2105	3647	3727
Max #	29946	29781	74588	222928

spatial patterns of the measurements are due to the sensor boxes being mounted on tram cars.

2.2. LAEI data

For showcasing the global applicability of our approach and the potential of globally available features, apply our features on annual mean PM_{10} concentrations stemming from the LAEI dataset (Greater London Authority, 2016). The data was obtained from a detailed dispersion model based on a vast number of input factors like road and rail traffic, aviation, agriculture, industry and domestic and commercial fuel burning and fires. For the dataset generation we randomly sampled 3000 datapoints for training and 1500 for testing purposes from the urban central London region. With this comparatively large dataset we are able to provide evaluation scores that are robust against outliers.

Table 2 shows the mean and standard deviation (SD) of both training and testing dataset. Note that the mean concentrations are higher than in Table 1, as PM_{10} includes bigger particles on top of UFP.

3. The OpenLUR approach

In this section, we introduce the main components of OpenLUR, our off-the-shelf approach for building air quality models based on land use regression (LUR): a novel set of open and globally available features derived from OpenStreetMap as well as the concept of automated hyper-parameter tuning for state-of-the art machine learning methods.

3.1. OpenStreetMap features

In contrast to feature sets used in previous studies (Hasenfratz et al., 2014; Aguilera et al., 2007; Briggs et al., 2000), our features are only based on OSM and thus are openly available and globally applicable.

To assess the validity of our feature set we compare them to a set of features used in previous work. In particular, we focus on the features from Hasenfratz et al. (2014).

Hasenfratz et al. (2014) derived features for each individual grid cell (cf. Section 2), including for example population or industry density, building heights or terrain properties shown in Table 3. While some of these features are derived from OpenStreetMap, most of them stem from data provided by governmental institutions in Switzerland and Zurich. Thus, they are only available in this region, which leads to a model, that is only applicable in Zurich and can not be compared to models designed fo other regions.

Table 2
Statistics for the LAEI PM_{10} dataset from London.

Dataset	Size	Mean $[\frac{10^9 particles}{m^3}]$	SD $[\frac{10^9 particles}{m^3}]$
Training	3000	28.15	2.78
Testing	1500	28.13	2.43

Table 3
Baseline features from Hasenfratz et al. (2014) with their respective source. Except from OSM, none of the features are globally available.

Feature	Source
Population density	Swiss Federal Statistical Office
Industry density	Swiss Federal Statistical Office
Building heights	Swiss Federal Statistical Office
Heating type	Swiss Federal Statistical Office
Terrain elevation	Swiss Federal Statistical Office
Terrain slope	Swiss Federal Statistical Office
Terrain aspect	Swiss Federal Statistical Office
Road type	OSM
Distance to next road	OSM
Distance to next large road	OSM
Distance to next traffic signal	OSM
Average daily traffic volume	Department of Waste, Water, Energy and Air of the Canton of Zürich

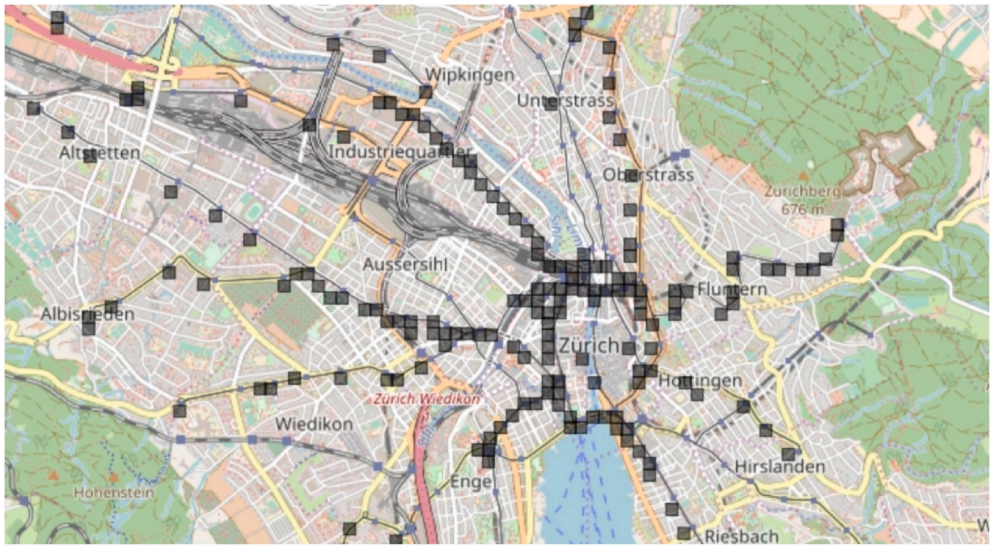


Fig. 2. An excerpt from the air pollution data from Zurich used for training LUR models. The figure shows the spatial distribution of the data from Season 2 of the OpenSense dataset (Hasenfratz et al., 2014) collected via sensor boxes on trams. The individual measurements are aggregated based on $100m \times 100m$ grid cells. Analogously to the experiments in (Hasenfratz et al., 2014), the cells are restricted to those 200 with the most measurements. ©OpenStreetMap contributors (www.openstreetmap.org/copyright).

To derive our novel set of globally available features, we employ OpenStreetMap (OSM) which provides openly and globally available land use data. In this section, we briefly introduce OSM as a data source and describe the features as well as their extraction process.

3.1.1. OpenStreetMap

OSM is an open source map dataset developed and maintained by a large number of volunteers from all around the world (Haklay and Weber, 2008). Many studies confirm the quality of the data provided by OSM (Haklay, 2010; Hecht et al., 2013). Consequently, OSM is a popular data source in a variety of studies ranging from risk management (Schelhorn et al., 2014) and disaster warning (Rahman et al., 2012) to navigation (Hentschel and Wagner, 2010) and routing (Luxen and Vetter, 2011). OSM also contains many variables related to air pollution. For example, it lists *key:value* pairs like *landuse:industry* or *highway:motorway* which can be directly used to derive relevant land usage and land cover statistics (Heymann, 1994; Estima and Painho, 2015; Hasenfratz et al., 2014)..

3.1.2. Feature extraction

To extract air pollution related features, we rely on OSM entities which are stored as polygons, lines or points (such as buildings, streets or traffic lights, respectively). Each entity is associated with a set of *key:value* pairs. In this study, we focus on entities with the keys *landuse* and *highway*. Using entities with these keys, we extracted two types of features: *area/length-based features* and *distance-based features*. These features are generated for each grid cell individually. We provide an overview over the features in Fig. 3.

For the *area/length-based features*, we define a circular zone (buffer) of various sizes around a grid cell's center (see top left of Fig. 3). Within those buffers, we measure the overall area or the overall length covered by those OSM entities relevant to the specific feature. In particular, we derive three area-based features by summing up the areas of entities with the key *landuse* and the values *industrial*, *commercial* and *residential* respectively. For the *length-based features*, we define two categories: roads with heavy traffic and roads with light traffic. For the *heavy traffic features*, we sum up the length of entities with the key *highway* and the values *motorway*, *trunk*, *primary* and *secondary*. For the *light traffic features*, we sum up the length of entities with the key *highway* and the values *tertiary* and *residential*. This provides information about industrial land-use and traffic intensity. The procedure is illustrated in Fig. 3. We varied the buffer radii in 50*m*-steps ranging from 50*m* to 3000*m* for *area-based features* and from 50*m* to 1500*m* for *length-based features* to account for distance-dependencies. The radii were chosen according to their maximum distance of influence (Jerrett et al., 2004; Henderson et al., 2007; Su et al., 2009). Three area-related *key:value* pairs using 60 buffer radii each and two length-related *key:value* pairs with 30 buffer radii each result in 240 features.

For the *distance-based features*, we focus on the *key:value* pairs *highway:motorway*, *highway:primary*, *highway:traffic_signals* and *landuse:industrial*. For each of these pairs, we calculate the distance between a grid cell's center and the nearest occurrence of an entity with the respective pair as illustrated in Fig. 3. Like for the area-based features, this represents information on the local traffic profile as well as industrial factors which are assumed to negatively influence air quality. Considering the four mentioned *key:value* pairs, this results in 4 features.

Combining both feature classes results in 244 open and publicly available features derived solely from OSM, shown in Table 4. By construction, these features represent land cover and traffic related information and are closely tied to air pollution. We are aware, that this list of land use features is not exhaustive as factors like elevation, population density and other meteorological and environmental covariates can also highly influence air quality. OpenLUR can be extended with additional data sources via an easy to use API. The aim of this study however is to show the capability of OSM to provide land use information that can outperform closed source land use features.

Table 4
Features derived from OpenStreetMap. The features are divided into two classes: area/length (top part) and distance based features (bottom part), where area/length features use different buffer sizes (50m–3000m/50m–1500m with a step size of 50m). Overall this results in 244 features for each grid cell.

Variable	Unit	*key:value* pairs in OSM
Industry usage	Area [m^2]	*landuse:industrial*
Commercial usage	Area [m^2]	*landuse:commercial*
Residential usage	Area [m^2]	*landuse:residential*
Heavy traffic	Length [m]	*highway:motorway* *highway:trunk* *highway:primary* *highway:secondary*
Light traffic	Length [m]	*highway:tertiary* *highway:residential*
Distance to next motorway	Distance [m]	*highway:motorway*
Distance to next primary road	Distance [m]	*highway:primary*
Distance to next traffic signal	Distance [m]	*highway:traffic_signals*
Distance to next industrial area	Distance [m]	*landuse:industrial*

3.2. Automated hyper-parameter tuning

Newer advancements in machine learning often promise better prediction results using the same data. These models however commonly require tedious hyper-parameter tuning and expert knowledge concerning the applied algorithms. In this section we briefly introduce several approaches for automatic hyper-parameter tuning to negate this disadvantage. This is one key feature of our off-the-shelf approach.

3.2.1. Basic approaches

Most state-of-the-art machine learning methods need to be tailored to specific tasks by selecting an appropriate set of hyper-parameters. For the example of random forests, the number of estimators, the number of features per estimator or the minimal number of samples per leaf have to be tuned. The typical procedure to tune hyper-parameter sets is as follows: The dataset is split in a train, a validation and a test set. Different hyper-parameter sets are trained on the train data and tested on the validation data. The best performing model is used as final model, retrained on train and validation set and tested on the test set. Due to the combinatorial explosion of possible hyper-parameter combinations, this process either requires expert knowledge or has to be automated. In the following, we revisit two commonly used generic methods to automatically optimize hyper-parameters: grid and stochastic search.

3.2.1.1. Grid search. Grid search is performed by manually choosing a set of candidate values for each hyper-parameter. Then, all possible combinations of these values are evaluated.

3.2.1.2. Stochasic search. Stochastic search optimizes hyper-parameters by randomly choosing candidate values for each hyper-parameter from a predefined probability distribution (mostly uniform) within a given time budget. This often allows to "find better models by effectively searching a larger, less promising configuration space" (Bergstra and Bengio, 2012) than manual or grid search.

3.2.2. AutoML

Automated Machine Learning (AutoML) (Blum et al., 2015) goes one step further than the standard way to automated parameter tuning. It builds an ensemble learner that exploits the synergy of several weak regressors to produce an improved model. In other words, it simultaneously chooses and combines models from a set of model classes (random forests, support vector machines, naive Bayes, etc.) while *at the same time* optimizing their hyper-parameters. For this, it does not rely on

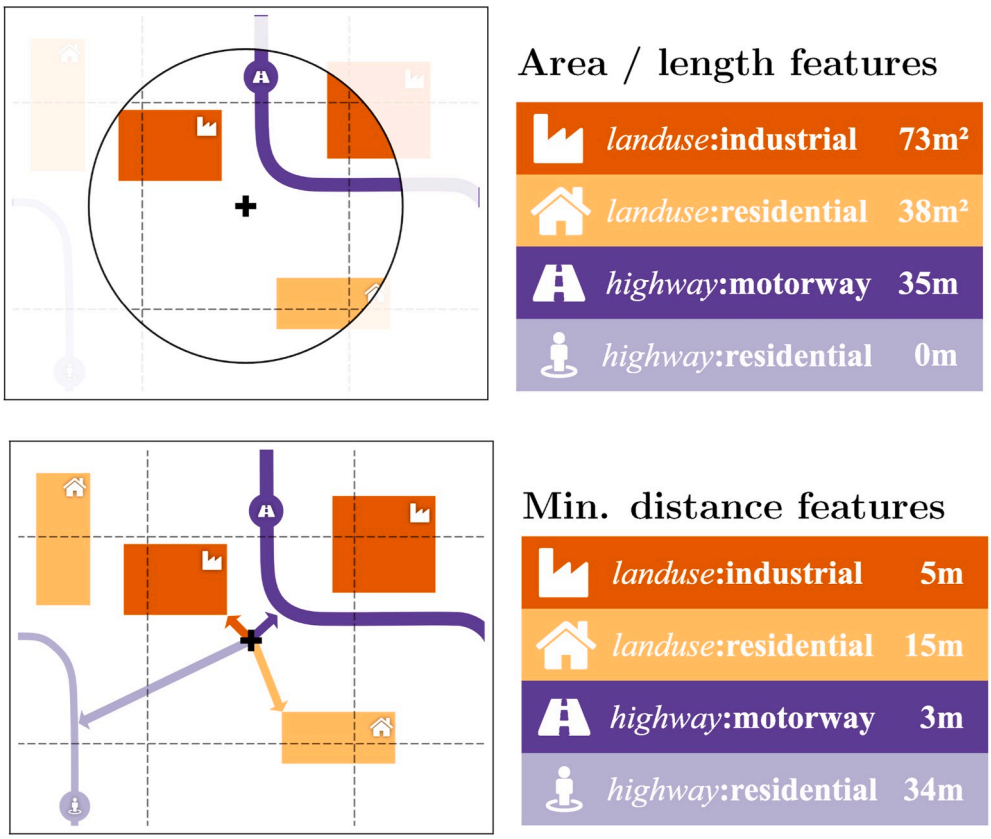


Fig. 3. Visualization of our set of open and globally available features. The top picture shows features based on the area/length of land use related entities within a given buffer zone. The bottom picture shows features based on the minimum distance to certain land use related entities.

grid or stochastic search, but utilizes efficient Bayesian optimization methods based on Gaussian processes to intelligently pick the most promising model and hyper-parameter combinations while staying within a given computational budget, such as time or memory usage (Blum et al., 2015).

4. Experimental setup

For OpenLUR we evaluate the two key components introduced in Section 3: our set of OSM features and the concept of hyper-parameter tuning for state-of-the-art machine learning methods. To show the ability of these components to provide an off-the-shelf approach, we compare our globally available OSM features with baseline features from previous work as independent variable and evaluate the competitiveness of machine learning methods featuring automated hyper-parameter tuning.

In this context, our general experimental setup is as follows: We aim to train models to predict the dependent variable UFP concentration at unobserved locations for the four seasons of the OpenSense dataset listed in Section 2. As done in most previous work, R^2 and the root mean squared error (RMSE) are computed as scores to compare their performance. The spatial dependence of the UFP concentration is modelled through the spatial variations of the independent variables. The temporal dependence is ruled out by averaging measurements over seasons (c.f. Section 2) and using only one season for each model building and evaluation process. To account for random outliers of these scores due to the inherently small training sets ($\leq$ 200 labeled samples in the OpenSense datasets, cf. Section 2), we report the mean of 40 10-fold cross validation scores as the final score for each model (cf. Hasenfratz et al. (2014)): For each of the 40 iterations, the dataset is randomly split into 10 subsets. Ensuingly each subset is used once for the evaluation while the models are built based on the 9 remaining subsets.

As baselines we picked two models that have proven to perform good on state-of-the art land use regression tasks (Hasenfratz et al., 2014; Champendal et al., 2014; Brokamp et al., 2017). To evaluate our apporach we compare them against two machine learning methods featuring automated hyper-parameter tuning. This results in the following list of models:

- GAM: generalized additive model (no hyper-parameter tuning) (Hastie and Tibshirani, 1986)
- RF: random forest (no hyper-parameter tuning) (Breiman, 2001)
- RFOstochastic: random forest (hyper-parameters tuned by stochastic search) (Breiman, 2001; Bergstra and Bengio, 2012)
- AutoML: automated machine learning (automated hyper-parameter tuning) (Blum et al., 2015)

Due to the technical limitations of GAMs, a small set of features needs to be selected. We explain this process in the supplementary material. Beyond evaluating different feature sets, we compare GAMs and untuned random forests, against two state-of-the-art models with automated hyper-parameter tuning. Besides AutoML, we chose to optimize random forests using stochastic hyper-parameter search since (i) random forests are one of the most popular machine learning methods for land use regression (Champendal et al., 2014; Brokamp et al., 2017; Araki et al., 2018; Stafoggia et al., 2019) and (ii) stochastic search is reported to outperform manual or grid search (Bergstra and Bengio, 2012).

Note that, the features we extracted from OpenStreetMap as well as the code used to produce the following results are publicly available at https://www.dmir.org/OpenLUR. A more detailed explanation of the experimental setup can be found in the supplementary material.

5. Results

In this section, we report the results based on the experimental setup described in Section 4. This encompasses (i) results on comparing our novel OSM features against a baseline feature set, (ii) results on comparing machine learning methods with and without hyper-parameter tuning, (iii) the application of OpenLUR on the LAEI dataset to evaluate the number of data samples needed for competitive results and the applicability of cross-learning across different cities, namely Zurich and London, to overcome the limits of small-scale air quality datasets, and (iv) a summary of the results and a recommendation of the overall approach for OpenLUR.

5.1. Feature comparison

In this section, we evaluate the performance of our feature set introduced in Section 3.1 and compare it with specialized — however only locally available — features from previous work (Hasenfratz et al., 2014). For this, we train several air quality models using both sets of features on the four seasons introduced in Section 2.

As a measure of absolute performance gain, we calculated the performance difference of our novel feature set and the OpenSense features for each model and each season with regard to RMSE and R^2, respectively. Table 5 shows the results. For RMSE (R^2) negative values (positive values) indicate a better model performance in favor of our proposed OSM features. Bold values indicate a statistically significant difference using the Wilcoxon signed-rank test ($p < 0.05$ with p values are provided in the supplementary materials).

We observe that in nearly all cases, our novel features yield significantly better results compared to the baseline features. That is, out of the 32 differences, 24 show a significant improvement in model quality. Only in four cases there is a significant tendency towards the baseline features. For the remaining cases our features perform equally well. The latter cases focus on Season 3 pointing towards a very specific data configuration that does not seem to be representative across the evaluated datasets (e.g., due to significant temperature drifts from October to December).

Thus, our novel and open OSM features significantly improve the performance of all studied air quality models compared to specialized and possibly restricted data sources. The globally available features enable air quality models to be trained anywhere on earth where OSM data is available.

5.2. Model comparison

We further evaluate the potential of automated hyper-parameter tuning. We therefore focus on our OSM features since they promise to yield the overall best results. The results for the baseline features are listed in the supplementary material. The results in Table 6 show the performance of each model listed in Section 4 for each season with regard to RMSE and R^2. The models are also ranked from best (1) to worst (4) on their performance in each Season. To facilitate an overall comparison between the models, the table furthermore lists the mean rank across all seasons.

We do not observe statistically significant differences between the regression models. Nevertheless, examining the mean ranks as an alternative evaluation measure, we clearly observe a tendency for models with automated hyper-parameter tuning to perform better than regular models. This holds with regard to both metrics and confirms that hyper-parameter tuning is an essential step for training land use regression models for air quality prediction.

Of those methods featuring automated hyper-parameter tuning, RFOstochastic performs better on both metrics. We assume that due to our rather small dataset the AutoML approach based on Gaussian processes can not exploit its full potential. On top of that Blum et al. (2015) showed, that the AutoML model needs a considerable amount of time to be able to outperform competitors like random forest (Blum et al., 2015). However, we expect the AutoML-based methods to show their advantage on larger datasets, where training models is more expensive and selecting particularly promising sets of hyper-parameters is essential. This needs to be further investigated.

Nevertheless, we have shown that advanced machine learning approaches employing automated hyper-parameter tuning, i.e., AutoML and RFOstochastic, are applicable to air pollution modeling and outperform the baseline methods when considering mean ranks across several experiments. We thus have shown that methods employing hyper-parameter tuning can alleviate manual effort while not compromising on prediction quality.

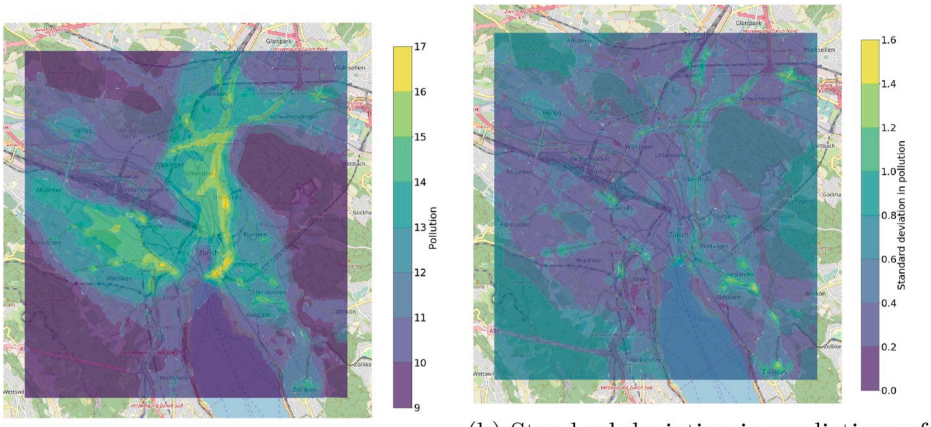
Fig. 4(a) depicts the spatial distribution of UFP in Zurich predicted by the RFOstochastic trained on OpenSense season 1 with our OSM features. Some patterns are clearly distinguishable: The water area as well as the recreation area in the western central part of the predictions are less polluted than urban Zurich. The higher pollution along some major roads is also visible. Fig. 4(b) shows the standard deviation of multiple resampled OpenLUR runs. The standard deviations are low with values up to 1.6 $[\frac{10^9 particles}{m^3}]$ while absolute predictions are values of up to 17 $[\frac{10^9 particles}{m^3}]$. The deviations are equally low in areas with lower or higher

Table 5
The absolute performance gain of our OSM based features over the OpenSense features with regard to RMSE and R^2 is shown. Negative (positive) RMSE (R^2) values show a better model performance when using our openly available OSM features and are highlighted in gray. In nearly all cases, our OSM features yield significantly better air quality predictions (bold values indicate statistical significance).

Season	AutoML	RFOstochastic	RF	GAM
	RMSE $[\frac{10^9 particles}{m^3}]$			
1	**−0.18**	**−0.24**	**−0.24**	**−0.30**
2	**−0.06**	**−0.08**	**−0.07**	**−0.07**
3	−0.05	0.06	**0.06**	**0.10**
4	**−0.34**	**−0.39**	**−0.40**	**−0.19**
	R^2			
1	**0.11**	**0.16**	**0.16**	**0.18**
2	**0.04**	**0.05**	**0.06**	**0.04**
3	0.03	−0.02	**−0.06**	**−0.06**
4	**0.13**	**0.19**	**0.19**	**0.03**

Table 6
RMSE and R^2 metrics of the models using OSM features. Parenthesis show the rank of the model given a particular season for the corresponding metric. Generally, the model performances do not differ significantly. However, a clear tendency towards models featuring automatic hyper-parameter tuning can be observed judging by their mean rank over all seasons.

Season	AutoML	RFOstochastic	RF	GAM
	RMSE $[\frac{10^9 particles}{m^3}]$ (rank)			
1	2.06 (3)	2.01 (2)	2.12 (4)	2.00 (1)
2	1.75 (2)	1.74 (1)	1.82 (4)	1.75 (2)
3	2.87 (1)	2.91 (2)	3.07 (3)	3.13 (4)
4	3.55 (1)	3.55 (1)	3.69 (3)	3.73 (4)
Mean rank	1.75	1.50	3.50	2.75
	R^2 (rank)			
1	0.40 (3)	0.43 (1)	0.36 (4)	0.42 (2)
2	0.38 (2)	0.39 (1)	0.32 (4)	0.37 (3)
3	0.35 (1)	0.32 (2)	0.25 (3)	0.21 (4)
4	0.19 (1)	0.19 (1)	0.11 (3)	0.08 (4)
Mean rank	1.75	1.25	3.50	3.25



(a) Predictions of UFP. (b) Standard deviation in predictions of

Fig. 4. Predictions and standard deviations of OpenLUR trained on season 1 of the OpenSense data with values in $[\frac{10^9 particles}{m^3}]$. ©OpenStreetMap contributors (www.openstreetmap.org/copyright).

pollution, best visible in the western, higher polluted, region of Zurich. This means that the resampled models have a high agreement on the predictions, independent of the predicted pollution value.

To show that the spatial dependence of the pollutant is modelled by the OSM features, as stated in Section 4, we computed the spatial correlation of the residuals with each independent variable. The correlation coefficient c is between -0.09 and 0.19. The coefficient of determination c^2 describes the percentage of the residuals that can be explained by the independent variable (Taylor, 1990). With $c^2 < 4\%$ the residuals are independent from our OSM features, meaning that the spatial dependence of UFP is indeed modelled by the independent variables.

In conclusion, the predictions achieved with OpenLUR are of a competitive prediction quality (c.f. Tables 5 and 6) and subjectively reasonable (c.f. Fig. 4(a)).

5.3. Global applicability and cross-learning

In this section, we utilize the LAEI dataset to show the applicability and potential of globally available land use features. We will first apply OpenLUR on the LAEI dataset and second demonstrate how the global applicability of our features can be used to improve the performance of LUR for small datasets through cross learning.

We use our OSM features and the best working models from the previous experiment, namely AutoML and RFOstochastic.

To show the global applicability of OpenLUR on datasets of different sizes, we first apply OpenLUR to different subsets (from 10 up to 1000 datapoints) of the LAEI data. The results are shown in Table 7. While the results of the models do not differ significantly, the accuracy rises with the size of the training data with an especially strong increase below 200 training samples. For a high number of training samples (≥ 300), AutoML seems to slightly ($\Delta R^2 \approx 0.01$), though not significantly, outperform RFOstochastic.

Datasets in LUR scenarios however are mostly small (< 200 data points). We can exploit the potential of our globally available OSM features through cross-learning on multiple cities to overcome the limit of small datasets and enhance the model accuracy. For cross-learning we increase the size of our training dataset by adding data samples from another region. In our case, we added 180 OpenSense data points from season 1 to the LAEI training samples.

Since both datasets measure different pollutants (PM_{10} and UFP respectively) the concentration values are in different ranges (see Section 2). To just exploit the similar dependence of both pollutants on land use features, we standardized the measurements of both datasets to a mean of 0 and a standard deviation of 1. To retrieve pollutant predictions, the output of the resulting model has to be transformed back using the mean and standard deviation of the original dataset.

The performance gain is shown in Table 8, where a positive value indicates a better performance through cross-learning. For small datasets, the enhancement of R^2 is significant (up to a LAEI dataset size of 40 for AutoML and 60 for RFOstochastic) with $p < 0.05$ (p-values are shown in the supplementary material). For bigger LAEI subsets, the performance gain is small and not significant. Especially for small datasets, cross-learning on multiple cities provides an opportunity to improve the model performance. Interestingly, for AutoML there is also a significant, however small ($\Delta R^2 \approx 0.01$), improvement for 1000 LAEI samples. The model enhancement through cross-learning on two cities gives a glimpse of the potential of our globally available features.

Table 7
$R^2$of both models, RFOstochastic and AutoML, applied on subsets of the LAEI dataset. This shows, that OpenLUR performs well, when a sufficient amount of training data is available.

Number of LAEI samples	AutoML	RFOstochastic
20	−0.08	0.01
40	0.21	0.13
60	0.29	0.27
80	0.39	0.41
100	0.42	0.41
150	0.48	0.49
200	0.51	0.51
300	0.54	0.53
400	0.58	0.57
500	0.59	0.58
1000	0.63	0.63

Table 8
R^2 performance gain (difference of the R^2 of models trained on both datasets and the R^2 of models trained exclusively on LAEI data). Positive values show a performance gain through cross learning. Especially small datasets can significantly take advantage of cross learning.

Number of LAEI samples	AutoML	RFOstochastic
20	0.29	0.21
40	0.10	0.16
60	0.05	**0.10**
80	0.04	0.00
100	0.02	0.02
150	0.02	0.00
200	−0.01	0.00
300	0.01	0.00
400	0.00	0.01
500	0.00	0.01
1000	**0.01**	0.01

5.4. Recommended off-the-shelf approach for predicting air pollution

For our OpenLUR approach, we recommend a combination of our novel OSM features and the AutoML model. With regard to the features, this recommendation is justified by the fact that the OSM features significantly outperformed the baseline features (Hasenfratz et al., 2014) on the given dataset and have the advantages of being openly and globally available. The global availability enables land use regression even for small datasets through cross-learning. With regard to the underlying model, the choice is less clear. While stochastically optimized random forests have a slight advantage with regard to the mean rank, we still recommend AutoML for its more sophisticated hyper-parameter tuning based on Gaussian processes which seems to yield a better prediction performance on larger datasets. The gobal applicability of OpenLUR facilitates through cross-learning LUR studies on small datasets and could furthermore be used for multi-city or even global scale LUR research. Thus, with its novel openly and globally available feature set in combination with the notion of automated hyper-parameter tuning to eliminate tedious parameter studies, OpenLUR provides a reproducible, easily and widely applicable off-the-shelf land-use regression approach for air quality prediction even for small datasets that does not require expert knowledge in machine learning.

6. Future work and implications

In this section, we discuss several directions of future work as well as important implications of OpenLUR. In particular, (i) we discuss further potential features, approaches and models, (ii) we list some limitations of our dataset, and (iii) review the possibilities and limitations of globally available land use regression features.

6.1. Features and machine learning methods

While we have introduced an air quality regression pipeline that outperforms previously proposed methods, our novel features as well as our applied models can be further refined and extended:

First, there is an endless amount of features to incorporate into air quality models: For example, based on OSM data, information about crossings, parks or specific venues like shops or sights has not been explored yet. Also, besides static land usage features, traffic models (Krauβ, 1998; Smith, 1993), open weather data or wind flow models could account for time-dependencies. However, the openly available code of OpenLUR provides the possibility to add custom features without restrictions to type or origin.

Second, other machine learning algorithms than those covered in this work can be considered. In the supplementary material, we present some additional experiments. Concretely, we optimize random forests with auto-sklearn using Bayesian as well as stochastic optimization in combination with ensemble learning, but none was able to consistently outperform AutoML and RFOstochastic. Additionally, we evaluated two more recently applied methods as baselines: geographically weighted regression (Alam and McNabola, 2015) and feed forward neural networks (Hu et al., 2013) (results in the supplementary materials). However, with the limited amount of 200 datapoints, these models did not perform well. Nevertheless, models based on neural networks may be interesting to explore, as they may be able to alleviate the issue of deriving specific features from the OSM attributes by directly providing raw OSM data.

Finally a disadvantage of most nonlinear state-of-the-art machine learning models as used in OpenLUR is the more difficult interpretability: Unlike simple linear regression models, the influence of an independent variable is not measured by a single value — the respective slope — but is hidden in more complex model structures. Research to interpret these models has been conducted (Palczewska et al., 2014; Fabris et al., 2018). This is however out of the scope for this study and will be treated in future work.

6.2. Dataset and measurements

A crucial point for developing air quality regression models is the quality and quantity of measurement data. While traditional studies used stationary monitoring devices that resulted in a small number of datapoints (Montagne et al., 2015; Ragettli et al., 2014), recent studies show the potential of large amounts of mobile measurements. Mobile devices however are usually prone to inaccurate measurements and noise. To counteract short-term disturbances like bypassing trucks or simply wind, the measurements mostly get aggregated temporally or spatially which, analogously to the static case, results in fewer datapoints. Nevertheless, at least for our dataset, the spatial coverage was still a lot larger than using a handful of static devices. In future studies, it may be of interest to directly compare the quality of continuous mobile measurements with static approaches. This will require potentially very expensive, large scale measurement campaigns.

6.3. Towards global land use regression models

Because of the openly and globally available features, the models produced by OpenLUR can be applied in any city with comparable OSM data. With the LAEI data we have shown the global applicability as well as the ability for cross-learning.

But locally differing characteristics of cities or the underlying OSM data (Davidovic et al., 2016), e.g., caused by structural difference of cities in different countries, conceptually differing ways of providing data within local OSM communities, or the general quality of the provided information, can lead to different dependencies of pollutants on the features.

This points to the scientifically highly interesting area of the effects of local air pollution environments as well as the characteristics of area-specific OSM data. Neverthless, our results are promising and present an important step towards generalized global air pollution models.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Florian Lautenschlager: Conceptualization, Methodology, Software, Formal analysis, Writing - original draft. **Martin Becker:** Writing - review & editing, Conceptualization, Methodology, Funding acquisition, Supervision. **Konstantin Kobs:** Writing - review & editing, Visualization. **Michael Steininger:** Writing - review & editing. **Padraig Davidson:** Writing - review & editing. **Anna Krause:** Writing - review & editing, Supervision. **Andreas Hotho:** Writing - review & editing, Conceptualization, Funding acquisition, Supervision.

Acknowledgements

This work has been partially funded by the DFG grant “p2Map: Learning Environmental Maps - Integrating Participatory Sensing and Human Perception” (Grant-nr: 314699772).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.atmosenv.2020.117535.

Funding

This work was partially funded by the DFG project p2map.

References

Aguilera, I., Sunyer, J., Fernández-Patier, R., Hoek, G., Aguirre-Alfaro, A., Meliefste, K., Bomboi-Mingarro, M.T., Nieuwenhuijsen, M.J., Herce-Garraleta, D., Brunekreef, B., 2007. Estimation of outdoor no x, no2, and btex exposure in a cohort of pregnant women using land use regression modeling. Environ. Sci. Technol. 42, 815–821.

Alam, M.S., McNabola, A., 2015. Exploring the modeling of spatiotemporal variations in ambient air pollution within the land use regression framework: estimation of pm10 concentrations on a daily basis. J. Air Waste Manag. Assoc. 65, 628–640.

Arain, M., Blair, R., Finkelstein, N., Brook, J., Sahsuvaroglu, T., Beckerman, B., Zhang, L., Jerrett, M., 2007. The use of wind fields in a land use regression model to predict air pollution concentrations for health exposure studies. Atmos. Environ. 41, 3453–3464. https://doi.org/10.1016/j.atmosenv.2006.11.063. http://www.sciencedirect.com/science/article/pii/S1352231006013008.

Araki, S., Shima, M., Yamamoto, K., 2018. Spatiotemporal land use random forest model for estimating metropolitan no2 exposure in Japan. Sci. Total Environ. 634, 1269–1277. https://doi.org/10.1016/j.scitotenv.2018.03.324. http://www.sciencedirect.com/science/article/pii/S0048969718310842.

Basu, B., Alam, M.S., Ghosh, B., Gill, L., McNabola, A., 2019. Augmenting limited background monitoring data for improved performance in land use regression modelling: using support vector regression and mobile monitoring. Atmos. Environ. 201, 310–322. https://doi.org/10.1016/j.atmosenv.2018.12.048. http://www.sciencedirect.com/science/article/pii/S1352231019300159.

Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. J. Mach. Learn. Res. 13, 281–305. http://www.jmlr.org/papers/v13/bergstra12a.html.

Blum, M., Feurer, M., Klein, A., Springenberg, J., Hutter, F., Eggensperger, K., 2015. Efficient and Robust Automated Machine Learning. NIPS, pp. 2962–2970. http://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning.

Brauer, M., Hoek, G., van Vliet, P., Meliefste, K., Fischer, P., Gehring, U., Heinrich, J., Cyrys, J., Bellander, T., Lewne, M., Brunekreef, B., 2003. Estimating long-term average particulate air pollution concentrations: application of traffic indicators and geographic information systems. Epidemiology 14, 228–239. https://doi.org/10.1097/01.EDE.0000041910.49046.9B.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32. https://doi.org/10.1023/A:1010933404324 doi:10.1023/A:1010933404324.

Briggs, D.J., de Hoogh, C., Gulliver, J., Wills, J., Elliott, P., Kingham, S., Smallbone, K., 2000. A regression-based method for mapping traffic-related air pollution: application and testing in four contrasting urban environments. Sci. Total Environ. 253, 151–167. https://doi.org/10.1016/S0048-9697(00)00429-0. http://www.sciencedirect.com/science/article/pii/S0048969700004290.

Brokamp, C., Jandarov, R., Rao, M., LeMasters, G., Ryan, P., 2017. Exposure assessment models for elemental components of particulate matter in an urban environment: a comparison of regression and random forest approaches. Atmos. Environ. 151, 1–11.

Brook, R.D., Rajagopalan, S., Pope, C.A., Brook, J.R., Bhatnagar, A., Diez-Roux, A.V., Holguin, F., Hong, Y., Luepker, R.V., Mittleman, M.A., 2010. Particulate matter air pollution and cardiovascular disease. Circulation 121, 2331–2378.

Carr, D., von Ehrenstein, O., Weiland, S., Wagner, C., Wellie, O., Nicolai, T., von Mutius, E., 2002. Modeling annual benzene, toluene, no2, and soot concentrations on the basis of road traffic characteristics. Environ. Res. 90, 111–118. https://doi.org/10.1006/enrs.2002.4393. http://www.sciencedirect.com/science/article/pii/S0013935102943938.

Champendal, A., Kanevski, M., Huguenot, P.E., 2014. Air pollution mapping using nonlinear land use regression models. In: International Conference on Computational Science and its Applications. Springer, pp. 682–690.

Davidovic, N., Mooney, P., Stoimenov, L., Minghini, M., 2016. Tagging in volunteered geographic information: an analysis of tagging practices for cities and urban regions in openstreetmap. ISPRS Int. J. Geo-Inf. 5, 232.

Dons, E., Poppel, M.V., Kochan, B., Wets, G., Panis, L.I., 2013. Modeling temporal and spatial variability of traffic-related air pollution: hourly land use regression models for black carbon. Atmos. Environ. 74, 237–246. https://doi.org/10.1016/j.atmosenv.2013.03.050. http://www.sciencedirect.com/science/article/pii/S1352231013002264.

Estima, J., Painho, M., 2015. Investigating the potential of openstreetmap for land use/land cover production: a case study for continental Portugal. In: OpenStreetMap in GIScience. Springer, pp. 273–293.

Fabris, F., Doherty, A., Palmer, D., De Magalhães, J.P., Freitas, A.A., 2018. A new approach for interpreting random forest models and its application to the biology of ageing. Bioinformatics 34, 2449–2456.

Greater London Authority, A.Q.T., 2016. London atmospheric emissions inventory (laei) 2013. https://data.london.gov.uk/dataset/london-atmospheric-emissions-inventory-2013.

Haklay, M., 2010. How good is volunteered geographical information? a comparative study of openstreetmap and ordnance survey datasets. Environ. Plann. Plann. Des. 37, 682–703.

Haklay, M., Weber, P., 2008. Openstreetmap: user-generated street maps. IEEE Pervasive Computing 7, 12–18.

Hankey, S., Marshall, J.D., 2015. Land use regression models of on-road particulate air pollution (particle number, black carbon, pm2.5, particle size) using mobile monitoring. Environ. Sci. Technol. 49, 9194–9202. https://doi.org/10.1021/acs.est.5b01209 pMID: 26134458.

Hasenfratz, D., Saukh, O., Walser, C., Hueglin, C., Fierz, M., Thiele, L., 2014. Pushing the spatio-temporal resolution limit of urban air pollution maps. In: 2014 IEEE International Conference on Pervasive Computing and Communications. PerCom, pp. 69–77. https://doi.org/10.1109/PerCom.2014.6813946.

Hastie, T., Tibshirani, R., 1986. Generalized additive models. Stat. Sci. 1, 297–310. https://doi.org/10.1214/ss/1177013604 doi:10.1214/ss/1177013604.

Hecht, R., Kunze, C., Hahmann, S., 2013. Measuring completeness of building footprints in openstreetmap over space and time. ISPRS Int. J. Geo-Inf. 2, 1066–1091.

Henderson, S.B., Beckerman, B., Jerrett, M., Brauer, M., 2007. Application of land use regression to estimate long-term concentrations of traffic-related nitrogen oxides and fine particulate matter. Environ. Sci. Technol. 41, 2422–2428. https://doi.org/10.1021/es0606780 pMID: 17438795.

Hentschel, M., Wagner, B., 2010. Autonomous robot navigation based on openstreetmap geodata. In: Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, IEEE, pp. 1645–1650.

Heymann, Y., 1994. CORINE Land Cover: Technical Guide. Office for Official Publ. of the Europ. Communities.

Hoek, G., Meliefste, K., Brauer, M., van Vliet, P., Brunekreef, B., Fischer, P., Lebret, E., Cyrys, J., Gehring, U., Heinrich, A., 2001. Risk Assessment of Exposure to Traffic-Related Air Pollution for the Development of Inhalant Allergy, Asthma and Other Chronic Respiratory Conditions in Children (Trapca). IRAS University, Utrecht. Final Report.

Hu, X., Waller, L.A., Al-Hamdan, M.Z., Crosson, W.L., Estes Jr., M.G., Estes, S.M., Quattrochi, D.A., Sarnat, J.A., Liu, Y., 2013. Estimating ground-level pm2. 5 concentrations in the southeastern us using geographically weighted regression. Environ. Res. 121, 1–10.

Jerrett, M., Arain, A., Kanaroglou, P., Beckerman, B., Potoglou, D., Sahsuvaroglu, T., Morrison, J., Giovis, C., 2004. A review and evaluation of intraurban air pollution exposure models. J. Expo. Anal. Environ. Epidemiol. 15, 185–204. https://doi.org/10.1038/sj.jea.7500388.

Krauß, S., 1998. Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics (Ph.D. thesis).

Larson, T., Henderson, S.B., Brauer, M., 2009. Mobile monitoring of particle light absorption coefficient in an urban area as a basis for land use regression. Environ. Sci. Technol. 43, 4672–4678. https://doi.org/10.1021/es803068e pMID: 19673250.

Luxen, D., Vetter, C., 2011. Real-time routing with openstreetmap data. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, pp. 513–516.

Minet, L., Gehr, R., Hatzopoulou, M., 2017. Capturing the sensitivity of land-use regression models to short-term mobile monitoring campaigns using air pollution micro-sensors. Environ. Pollut. 230, 280–290. https://doi.org/10.1016/j.envpol.2017.06.071. http://www.sciencedirect.com/science/article/pii/S0269749117309612.

Montagne, D.R., Hoek, G., Klompmaker, J.O., Wang, M., Meliefste, K., Brunekreef, B., 2015. Land use regression models for ultrafine particles and black carbon based on short-term monitoring predict past spatial variation. Environ. Sci. Technol. 49, 8712–8720. https://doi.org/10.1021/es505791g pMID: 26079151.

Muttoo, S., Ramsay, L., Brunekreef, B., Beelen, R., Meliefste, K., Naidoo, R.N., 2018. Land use regression modelling estimating nitrogen oxides exposure in industrial south durban, South Africa. Sci. Total Environ. https://doi.org/10.1016/j.scitotenv.2017.07.278, 610-611, 1439 – 1447. http://www.sciencedirect.com/science/article/pii/S0048969717319988.

OpenStreetMap contributors, 2017. Planet dump retrieved from. https://planet.osm.org. https://www.openstreetmap.org.

Palczewska, A., Palczewski, J., Marchese Robinson, R., Neagu, D., 2014. Interpreting Random Forest Classification Models Using a Feature Contribution Method. Springer International Publishing, Cham, pp. 193–218. https://doi.org/10.1007/978-3-319-04717-1_9 doi:10.1007/978-3-319-04717-1_9.

Patton, A.P., Collins, C., Naumova, E.N., Zamore, W., Brugge, D., Durant, J.L., 2014. An hourly regression model for ultrafine particles in a near-highway urban area. Environ. Sci. Technol. 48, 3272–3280. https://doi.org/10.1021/es404838k pMID: 24559198.

Polichetti, G., Cocco, S., Spinali, A., Trimarco, V., Nunziata, A., 2009. Effects of particulate matter (pm10, pm2.5 and pm1) on the cardiovascular system. Toxicology 261, 1–8. https://doi.org/10.1016/j.tox.2009.04.035. http://www.sciencedirect.com/science/article/pii/S0300483X09002121.

Pope, C.A., Dockery, D.W., Spengler, J.D., Raizenne, M.E., 1991. Respiratory health and pm10 pollution: a daily time series analysis. Am. Rev. Respir. Dis. 144, 668–674. https://doi.org/10.1164/ajrccm/144.3_Pt_1.668 doi:10.1164/ajrccm/144.3_Pt_1.668.

Ragettli, M.S., Ducret-Stich, R.E., Foraster, M., Morelli, X., Aguilera, I., Basagaña, X., Corradi, E., Ineichen, A., Tsai, M.Y., Probst-Hensch, N., Rivera, M., Slama, R., Künzli, N., Phuleria, H.C., 2014. Spatio-temporal variation of urban ultrafine particle number concentrations. Atmos. Environ. 96, 275–283. https://doi.org/10.1016/j.atmosenv.2014.07.049. http://www.sciencedirect.com/science/article/pii/S1352231014005792.

Rahman, K.M., Alam, T., Chowdhury, M., 2012. Location based early disaster warning and evacuation system on mobile phones using openstreetmap. In: Open Systems (ICOS), 2012 IEEE Conference on, IEEE, pp. 1–6.

Sahsuvaroglu, T., Arain, A., Kanaroglou, P., Finkelstein, N., Newbold, B., Jerrett, M., Beckerman, B., Brook, J., Finkelstein, M., Gilbert, N.L., 2006. A land use regression model for predicting ambient concentrations of nitrogen dioxide in Hamilton, ontario, Canada. J. Air Waste Manag. Assoc. 56, 1059–1069. URL:/brokenurl#. https://doi.org/10.1080/10473289.2006.10464542.

Schelhorn, S.J., Herfort, B., Leiner, R., Zipf, A., Albuquerque, J.P.d., 2014. Identifying elements at risk from openstreetmap: the case of flooding. In: International Conference on Information Systems for Crisis Response and Management, vol. 11. The Pennsylvania State University.

Shi, Y., Lau, K.K.L., Ng, E., 2016. Developing street-level pm2.5 and pm10 land use regression models in high-density Hong Kong with urban morphological factors.

Environ. Sci. Technol. 50, 8178–8187. https://doi.org/10.1021/acs.est.6b01807 pMID: 27381187.

Smith, M., 1993. A new dynamic traffic model and the existence and calculation of dynamic user equilibria on congested capacity-constrained road networks. Transp. Res. Part B Methodol. 27, 49–63.

Stafoggia, M., Bellander, T., Bucci, S., Davoli, M., de Hoogh, K., de' Donato, F., Gariazzo, C., Lyapustin, A., Michelozzi, P., Renzi, M., Scortichini, M., Shtein, A., Viegi, G., Kloog, I., Schwartz, J., 2019. Estimation of daily pm10 and pm2.5 concentrations in Italy, 2013–2015, using a spatiotemporal land-use random-forest model. Environ. Int. 124, 170–179. https://doi.org/10.1016/j.envint.2019.01.016. http://www.sciencedirect.com/science/article/pii/S0160412018327685.

Su, J., Jerrett, M., Beckerman, B., 2009. A distance-decay variable selection strategy for land use regression modeling of ambient air pollution exposures. Sci. Total Environ. 407, 3890–3898. https://doi.org/10.1016/j.scitotenv.2009.01.061. thematic Issue - BioMicroWorld Conference. http://www.sciencedirect.com/science/article/pii/S0048969709001442.

Su, J.G., Hopke, P.K., Tian, Y., Baldwin, N., Thurston, S.W., Evans, K., Rich, D.Q., 2015. Modeling particulate matter concentrations measured through mobile monitoring in a deletion/substitution/addition approach. Atmos. Environ. 122, 477–483. https://doi.org/10.1016/j.atmosenv.2015.10.002. http://www.sciencedirect.com/science/article/pii/S1352231015304210.

Sîrbu, A., Becker, M., Caminiti, S., De Baets, B., Elen, B., Francis, L., Gravino, P., Hotho, A., Ingarra, S., Loreto, V., Molino, A., Mueller, J., Peters, J., Ricchiuti, F., Saracino, F., Servedio, V.D.P., Stumme, G., Theunis, J., Tria, F., Van den Bossche, J., 2015. Participatory patterns in an international air quality monitoring initiative. PloS One 10, 1–19. https://doi.org/10.1371/journal.pone.0136763 doi:10.1371/journal.pone.0136763.

Taylor, R., 1990. Interpretation of the correlation coefficient: a basic review. J. Diagn. Med. Sonogr. 6, 35–39. https://doi.org/10.1177/875647939000600106 doi: 10.1177/875647939000600106.

Zwack, L.M., Paciorek, C.J., Spengler, J.D., Levy, J.I., 2011. Characterizing local traffic contributions to particulate air pollution in street canyons using mobile monitoring techniques. Atmos. Environ. 45, 2507–2514. https://doi.org/10.1016/j.atmosenv.2011.02.035. http://www.sciencedirect.com/science/article/pii/S1352231011001695.

Supporting Information: OpenLUR: Off-the-shelf air pollution modeling with open features and machine learning

Florian Lautenschlager,* Martin Becker, Konstantin Kobs, Michael Steininger, Padraig Davidson, Anna Krause, and Andreas Hotho

Chair of Data Science, Computer Science, University of Würzburg

E-mail: lautenschlager@informatik.uni-wuerzburg.de

In addition to the work presented in the main paper, this supplementary material covers an extensive related work section on air pollution studies on the one hand, and includes additional details and experiments on the other hand. The latter includes i) the feature selection procedure which we employed in order to apply generalized additive models in our experiments, ii) details on our experimental setup including, e.g., specific parameter settings, as well as iii) further results investigating several aspects of our work. The latter results include model performances with regard to the baseline OpenSense features (including the p-values for the comparison to our novel OSM-based features), experiments on additional models (e.g., geographically weighted regression or feed forward neural networks), and a small study on the optimization time for the stochastically optimized random forest and AutoML.

Related Work

In the past a lot of research regarding pollution measurement and modelling has been conducted. Here, we give an overview of different modes of measurements, i.e., stationary vs. mobile, and cover other air quality models besides land use regression.

Stationary measurements. Over time, many measurement campaigns as well as land use regression approaches have been studied.[S1–S33] Table S1 gives a comprehensive overview of these studies listing specific details like the measured variables, the number of measuring sites, the features for building the models, as well as the corresponding feature sources.

The listed papers all determined a measurement variable averaged over predefined timespans ranging up to several weeks to rule out noise as well as time dependencies. The most measured variables are NO2 (nitrogen dioxide) and PM2.5 (particulate matter up to 2.5 μm in diameter) but also NO (nitrogen monoxide), NOx (all reactive nitrogen oxides), UFP (ultrafine particles, PM0.1), VOC (volantile organic compound), black carbon (also known as soot and elemental carbon), BTEX (aromatic hydrocarbons) and absorbance of particulate matter have been measured.

The number of measurement sites ranged from 20 to 644, the latter, 644 sites, being an extreme outlier. The study with the second most sites only features 240 locations, distributed over three cities. The average number of sites per study is 93 distributed over one or more cities. Note, that the spatial sparseness of the measurement sites limits the information in the data about small scale variances, like the increase in pollution when moving towards a busy street.

After the measurement period, all the works tried to model sensed values dependent on surrounding conditions. Table S1 also shows the features and feature sources the different studies used. For most studies, these were mainly land usage, traffic and population density. They were mostly extracted from commercial software (i.e., from ESRI: ArcGIS, ArcView, ArcInfo, ...) and data from local or national governments. Five studies used data from CORINE (Coordination of Information on the Environment) that contains land use data on a 100 m grid, but is only available for the region of the EU.[S16,S23,S25,S27,S30] Some studies incorporated meteorological values such as wind fields and solar radiance into their models.[S12,S20,S21,S25,S32,S34,S35] This globally available data

can be obtained from commercial sources or national meteorological institutes.

While most measurement campaigns measured in periods of several weeks, the more recent ones tend to be short-term measurements,[S27–S29,S31,S33] where the sampling period ranges from seconds[S33] up to three hours.[S29] This shorter measurement duration results in the possibility to sense more sites in the same study duration compared to long-term measurements. On the other hand, this data may suffer from temporal variances, like rush-hour traffic.

While most studies use the time average of all measurements on one point as final air pollution value, some approaches average over a shorter time span, like hours, and incorporate this additional information in their models.[S30,S34] This is done by either using the time, in this example hours of the day, as input variables for the model or to compute different models for the different hours. In more recent research, time has also been modeled as a continuous input.[S35,S36]

Mobile measurements. In recent years, mobile measurement have become popular.[S32,S34–S39] More details are shown in Table S2 listing the same attributes as for the stationary studies: the measured variables, the number of measuring sites, the features for building the models, as well as the corresponding feature sources. Mobile measurements often suffer from noise, that is brought into the measurements by the movement: Even small movements, that are not detected by positioning systems like GPS, can have a lare effect on the actual measurements. Additionally, each spatial location is often only measured by a few recordings, which fails to measure temporal variations. Thus, while mobile measurements enhance the spatial density of the measurend variables, this often results in lower temporal coverage. In addition, to these issues mobile devices are often less accurate and result in more noisy air pollution readings than dedicated monitoring stations. To solve these flaws data is often averaged in space based on grid cells as we do in our study. While this does not solve the listed issues perfectly, it allows to treat the aggregated data like stationary measurements which we can model using land use regression as in the stationary case.

Models besides LUR. Other approaches try to predict the air quality based on other features, e.g., meteorological variables.[S40,S41] Meteorology has a big impact on transport and mixing of air pollutants, through winds, as well as decomposition, through solar radiance and precipitation,

thus strongly influencing the concentration of these pollutants. However, in contrast to LUR, the sources of the pollutants are not taken into account this way.

Also, there are studies building on mobile measurements, that try to predict air quality maps just based on the actual measurement and the geographic location.[S42] They are combining the high spatial resolution of mobile measurements and the better temporal resolution of some stationary monitoring sites. Since these models simply interpolate between measured points, they are only applicable, where measurements are available. Additionally, they do not take into account additional information like proven air pollutant sources, e.g., streets or factories.

Summary. All the abovementioned studies model stationary or aggregated mobile air pollution measurements. But depending on their features or modelling techniques, the resulting models use either commercial data, are only applicable in a restricted area, or do not model the spatial variability of the pollutants. Also, mostly only simple regression models have been used. In the main paper, we address these issues by proposing an off-the-shelf air pollution modeling pipeline based on openly available features.

Table S1: Stationary pollution measurements with land use models

Reference	**Measured variable**	**No. of sites**	**Features**	**Feature source**
Briggs et al.[S1]	NO2	240	traffic, land cover, altitude, sampling height	local gouvernmental data, field measurements
Stedman et al.[S2]	NO2, NOx	37	urban and suburban land cover, vehicle NOx emissions	Institute of Terrestrial Ecology,[S43] CORINE, national emission data
Briggs et al.[S3]	NO2	83	traffic, land cover, altitude, sampling height	GIS-data, local gouvernmental data, field measurements
Carr et al.[S4]	NO2, soot, VOC	34	traffic intensity, traffic jams	manually counted
Brauer et al.[S5], Hoek et al.[S44]	PM2.5, soot, NO2	122	population, household density, traffic, region	ArcInfo, local governmental data
Gilbert et al.[S6]	NO2	67	distance to highway, traffic counts, area of open space, population	ArcGIS
Kanaroglou et al.[S7]	NO2	100	distance to expressway, traffic, land cover, household density	ArcView

Gonzales et al.[S8]	NO2	20	distance to US-Mexican border, distance to highway, altitude	ArcGIS
Smith et al.[S9]	NO2, VOC	22	distance to US-Mexican border, distance to major road, traffic, population, distance to petroleum facility, altitude	ArcGIS
Rossa et al.[S10]	NO2	39	traffic, length of road, distance to pacific coast, land use, population	local and national government
Hochadel et al.[S11]	NO2, PM2, absorbance, PM2.5	40	traffic, number of buildings, distance to major roads	ArcView
Sahsuvaroglu et al.[S12]	NO2	107	physical geography, meteorology, population, roads, traffic	Commercial sources (DMTI Spatial Inc), measurements, local government
Beelen et al.[S13]	NO2, NO, soot	16-36	land use, traffic, population	ArcGIS, national government
Ross et al.[S14]	PM2.5	36-62	traffic, population, land use, national PM2.5 emissions	ArcGIS, local and national government
Ryan et al.[S15]	soot	24	land usage, altitude, traffic	ArcGIS, local government

Morgenstern et al. [S16]	NO2, PM2.5, absorbance	40	land use, traffic, population, household density	ArcGIS, CORINE, national an local government
Moore et al. [S17]	PM2.5	23	land use, population, traffic, physical geography	ArcGIS, local and national government
Madsen et al. [S18]	NO2, NO, NOx	80	land use, traffic, population, altitude	ArcGIS, national government
Henderson et al. [S19]	NO2, NO, PM2.5, absorbance	25-116	land use, traffic, road length, population	government
Jerrett et al. [S20]	NO2	95	land use, traffic, population, physical geography, meteorology	ArcGis, commercial source, local and national government
Arain et al. [S21]	NO2	105	wind fields, land use, traffic, phytical geography	ArcGIS, national government
Rosenlund et al. [S22]	NO2	70	traffic, population, altitude, distance to sea, emission of pollutants	ArcGIS, local government
Aguilera et al. [S23]	NO2, NO, BTEX	57	land use, traffic, population, physical geography	ArcGIS, CORINE, local government
Wheeler et al. [S24]	NO2, VOC	54	traffic, population, dwelling density, land use, emission of pollutants	ArcGIS

Gulliver et al. [S25]	PM10	52	land use, traffic, population, meteorology	ArcGIS, CORINE, commercial data (Ordnance Survey Landline)
Su et al. [S26]	NO2	100	physical geography, land use, population, traffic, distance to roads	-
Rivera et al. [S27]	UFP	644	traffic, land use, population, household density	counted, CORINE, local government
Abernethy et al. [S28]	UFP, NOx	116	land use, traffic, population	ArcGIS, national government
Saraswat et al. [S29]	UFP, PM2.5, soot	30-48	land use, road distance and length, population, background population	ArcGIS, local and national government
Dons et al. [S30]	soot	63	traffic, population, lan use	CORINE, local model Bellemans et al. [S45]
Ragettli et al. [S31]	UFP	60	traffic, building height, meteorology	Measurements, no information for GIS-data
Montagne et al. [S32]	UFP, soot	161	land use, population, traffic, meteorogy, season	no information
Kerckhoffs et al. [S33]	UFP, soot	161	traffic, land use, population, household density, airports	no information

Table S2: Mobile pollution measurements with land use models

Reference	**Measured variable**	**Features**	**Feature source**
Larson et al. [S36]	BC	traffic, landuse, population	government
Zwack et al. [S34]	UFP	meteorology, day of week, traffic, physical geography	ArcGIS, measurements
Patton et al. [S35]	UFP	meteorology, time, traffic, distance to combustion sources	no information
Hasenfratz et al. [S38]	UFP	Population, building height, traffic, physical geography	OpenStreetMap, national government
Montagne et al. [S32]	UFP, BC	meteorogy, land use, population, traffic, season	no information
Hankey and Marshall [S39]	Particle Number, BC, PM2.5, Particle size	traffic, land use, population, physical geography	no information

Feature selection for baseline methods

The feature extraction procedure described in the main paper results in a large number of features (244). However, many regression models like linear regression or generalized additive models (as we use as baselines in this work) can only handle a limited amount of features dependent on the training set size. Therefore the most promising features have to be selected before building the

final model.

Methodology

To find the best features, we apply a feature selection procedure similar to previous work:[S23,S30,S32] First we calculate the R^2 score for each independent feature based on a univariate GAM. Then the independent feature with the highest R^2 is added to the selected features. Next, we calculate the R^2 of multivariate GAMs one for each remaining feature in addition to the already selected features. Again the variable with the highest R^2 is added to the selected features. This is repeated as long as new features add at least 0.01 to the overall R^2 score. To account for dependencies of the features and the predicted variable in the training/test split, each R^2 score was calculated as an average over 5 times 10-fold cross validation.

Results

For the baseline features Hasenfratz et al.[S38] already conducted a feature selection. Therefore we only selected appropriate features for our novel open source feature set. The feature selection procedure determined five to seven different features for each season that add at least 0.01 to the R^2 of the final model. The selected independent variables are shown in Table S3. The most important feature seems to be the residential area in different buffers as one of them gets always chosen first. Other features that are selected for every season are the length of big streets in a smaller buffer ($50m$–$200m$) and the industrial area in different buffers. The commercial area on the other hand seems to have a rather minor effects as it is only selected twice. Overall this shows that our novel OSM-based features are meaningful for air quality prediction, as industry and big streets are expected to have an influence on air pollution.

Table S3: Results of the feature selection.

Season	Chosen features
1	residential1050m, bigStreet100m, residential1950m, residential600m, industrial2500m, distanceMotorway, distanceTrafficSignal
2	residential2000m, bigStreet100m, commercial2100m, industrial2450m, industrial2950m
3	residential1850m, industrial1850m, bigStreet50m, residential1250m, residential2350m, residential400m
4	residential1300m, commercial2950m, industrial1700m, bigStreet200m, industrial550m

Experimental Setup

We have outlined the experimental setup in the main paper. However, we have skipped the detailed parameter settings for brevity. In this section, we give a detailed description of the setup.

As already, mentioned, we trained four models on the four seasons from the OpenSense dataset for different feature sets. For this, we used repreated cross-fold validation in order To account for random outliers of the scores due to the inherently small training sets ($\leq$ 200 labeled samples). In particular, we report the mean of 40 10-fold cross validation scores similar to Hasenfratz et al.[S38]. R^2 and the root mean squared error (RMSE) are computed as scores to compare the model results. For the different models we use implementations in Python if not denoted otherwise. The parametrizations are described in the following:

- GAM: The GAM model was calculated by the R package *mgcv* with cubic splines. It requires feature selection but no hyper-parameter tuning.[S46]
- RF: RF is a random forest with standard hyper-parameters provided by the *sklearn* package:
  - n_estimators = 10: *the number of trees in the forest.*
  - criterion = MSE: *the function to measure the quality of a split. MSE is the mean squared error.*

  - max_depth = NONE: *the maximum depth of a tree. NONE means, that each tree is expanded as far as possible.*
  - max_features = auto: *the number of features to consider when looking for the best split; auto means to consider all features*
  - min_samples_leaf = 1: *the minimum number of samples required to be at a leaf node*
  - min_samples_split = 2: *the minimum number of samples required for a split.*
  - bootstrap = TRUE: *whether to use bootstrap for sampling.*

- RFOstochastic: RFOstochastic also represents a random forest regressor, but with hyper-parameters optimized using stochasic optimization: The following hyper-parameters are optimized over the given distributions:
  - n_estimators: *uniformly distributed from 1 to 1000*
  - max_features: *uniformly distributed from 0 to 1*
  - min_samples_leaf: *uniformly distributed from 1 to 100*
  - min_samples_split: *uniformly distributed from 2 to 20*
  - bootstrap: *random boolean*

  Generally, the 10-fold split provides a training and test dataset. However, for the hyper-parameter optimization, the we need to split each training set into a tuning and a validation set. The optimization algorithm is given a predefined time to optimize hyper-parameters on the tuning and validation data. Afterwards, the best hyper-parameter set is selected and the model is retrained on the whole training data and evaluated on the test data.

- AutoML: AutoML requires no explicit hyper-parameter set. It only expects a fixed computational budget or a fixed time for the internal hyper-parameter search. As implementation we use the *auto-sklearn*[S47] package with disabled preprocessing.

AutoML as well as the RFOstochastic require predefined time intervals for the hyper-parameter search. In the main paper, we are presenting the results based on a tuning time of 300 seconds. In the "Optimization Time" section, we also report results on different intervals ranging from one to ten minutes; however, no significant impact on the results was found.

Additional Results

In addition to the results presented in the main paper, this section presents i) detailed results for our selected models using the baseline OpenSense features (inclduing p-values for the comparison to our novel OSM-based features), ii) experiments on additional models (e.g., feedforward neural networks), as well as iii) a study on the influence of optimization times for models featuring automated hyper-parameter tuning.

Baseline features (OpenSense)

In this section, we present the results of the models listed from the main paper but using the baseline OpenSense features, as well as the p-values for the comparison to our novel OSM-based feature set.

Model performance

In the main paper, we used the features presented by Hasenfratz et al.[S38] as baseline features to compare our novel OSM-based features against. For these baseline features, analogously to the main paper, each model for each season with regard to RMSE and R^2. It also shows the rank of each model — 1 (best) through 4 (worst) — with regard to its performance in each season. To facilitate an overall comparison between the models, the table furthermore lists the mean rank across all seasons.

As for our OSM-based feature set, considering the mean rank, the optimized models perform better than the unoptimized models for both metrics. For RMSE, both, AutoML and RFOstochas-

Table S4: RMSE and R^2 metrics of the models using OpenSense features. Parenthesis show the rank of the model given a particular season for the corresponding metric. Bold values inidicate statistically significant differences to all other models trained on the same season. Generally, the model performances do not differ significantly. However, as for our novel OSM-based features, a clear tendency towards models featuring automatic hyper-parameter tuning can be observed by judging their mean rank over all seasons.

Season	AutoML	RFOstochastic	RF	GAM
	RMSE (rank)			
1	2.25 (1)	2.25 (1)	2.36 (4)	2.30 (3)
2	1.81 (1)	1.81 (1)	1.89 (4)	1.82 (3)
3	2.94 (2)	2.86 (1)	2.95 (3)	**3.03** (4)
4	3.92 (1)	3.94 (2)	4.09 (4)	3.92 (3)
Mean rank	1.25	1.25	3.75	3.25
	R^2 (rank)			
1	0.28 (1)	0.27 (2)	0.21 (4)	0.25 (3)
2	0.35 (1)	0.34 (2)	**0.26** (4)	0.33 (3)
3	0.31 (2)	0.34 (1)	0.30 (3)	**0.27** (4)
4	0.03 (2)	-0.01 (3)	**-0.08** (4)	0.05 (1)
Mean rank	1.5	2.00	3.75	2.75

tic, are the best models for three seasons. Similar results can be observed for R^2, where AutoML can even outperform the RFOstochastic rank-wise. The RF and GAM are in two cases each even significantly worse than all other models, signaling the superiority of the optimized models. Also, AutoML achieves the best mean rank of 1.5, thus, outperforming RFOstochastic indicating a slight advantage of its underlying algorithm. Overall, the results are similar to those for our novel features reported in the main paper strengthening our recommendation for automated hyper-parameter tuning.

Feature Comparison - Significance

Table S5 shows the p-values obtained by the Wilcoxon signed-rank test for the comparion of the baseline OpenSense features and our novel OSM-based features (see main paper). As reported in the main paper, most of the values are below 0.05 and indicate significant differences.

Table S5: P-values of the Wilcoxon signed-rank test for the comparion of the baseline OpenSense features and our novel OSM-based features (see main paper). Bold numbers indicate values below 0.05.

Season	**AutoML**	**RFOstochastic**	**RF**	**GAM**
	RMSE			
1	**1.6e-08**	**1.1e-12**	**3.4e-13**	**5.4e-19**
2	**3.0e-02**	**3.8e-03**	**7.8e-03**	**4.6e-03**
3	9.3e-01	1.7e-01	**1.9e-03**	**9.9e-03**
4	**1.5e-08**	**2.6e-07**	**9.5e-07**	**1.4e-02**
	$\mathbf{R^2}$			
1	**5.8e-14**	**1.8e-20**	**1.7e-16**	**5.0e-23**
2	**1.5e-02**	**6.8e-04**	**2.9e-03**	**2.8e-03**
3	1.0e-01	1.1e-01	**2.6e-04**	**2.9e-04**
4	**2.5e-20**	**7.3e-17**	**5.6e-13**	**1.5e-04**

Additional Models

In addition to the models listed in the main paper, we also investigated different variants of automated hyper-parameter tuning as well as other models:

- **AutoMLstochastic** builds an ensemble the same way as AutoML, but uses stochastic hyper-parameter tuning instead of Bayesian processes.
- **RFObayes** optimizes a random forest using Bayesian processes. This is done using the *auto-sklearn* Python package.
- **RFOstochastic$_{\text{AutoML}}$** also optimizes the random forest using *auto-sklearn*. But instead of using Bayesian processes, a stochastic search is performed.
- **ANN** is an artificial neural network with one hidden layer consisting of 100 neurons, ReLU activation function and ADAM optimizer.
- **GWR** geographically weighted regression as described in Hu et al.[S48].

Table S6 shows the results of all models for a comparison. The results of the hyper-parameter tuning variants do not differ greatly. However, a few tendencies can be observed: AutoMLstochas-

Table S6: This table shows the RMSE and R^2 metrics of all tested models using our novel OSM-based features. The different hypre-parameter tuning variants do not differ significantly. The additional models (ANN, GWR) do not reach the performance of the other models probably due to missing hyper-parameter optimization and/or the limited amount of training data.

Season	**AutoML**	**AutoMLstochastic**	**RFOstochastic**	**RFObayes**	**RFOstochastic$_{\text{AutoML}}$**	**RF**	**GAM**	**ANN**	**GWR**
	RMSE								
1	2.06	2.07	2.01	2.14	2.08	2.12	2.00	1380.166	4.125
2	1.75	1.75	1.74	1.79	1.78	1.82	1.75	1607.332	3.552
3	2.87	2.93	2.91	2.95	2.94	3.07	3.13	1474.044	5.219
4	3.55	3.54	3.55	3.66	3.59	3.69	3.73	1275.188	5.313
	R^2								
1	0.40	0.40	0.43	0.34	0.38	0.36	0.42	-462110.673	-1.514
2	0.38	0.38	0.39	0.35	0.35	0.32	0.37	-710125.286	-1.6
3	0.35	0.32	0.32	0.30	0.32	0.25	0.21	-262609.661	-1.225
4	0.19	0.21	0.19	0.15	0.17	0.11	0.08	-180904.726	-0.903

tic to perform approximately equally well as the AutoML showing that the optimization mechanism used by AutoML based on Gaussian processes has not yet shown its full potential. We expect better results for larger datasets and signifantly longer runtimes.[S47] Also, a tendency for *auto-sklearn*-based optimization to perform worse than a direct implementation of stoachstic search can be observed. Especially the discrepancy of $\text{RFOstochastic}_{\text{AutoML}}$ and RFOstochastic seems counter-intuitive. The discrepancy could result from a less efficient implementation, where the algorithm tests less hyper-parameter combinations in the same time. However, to prove this, further investigation is required.

The new models, ANN and GWR, perform considerably worse than all other methods. This could be a consequence of both models' requirement for bigger training data quantity to build a generalizing model or the need for specific hyper-parameter tuning.

Overall, the results in this section confirm that automated hyper-parameter tuning can improve the quality of models and reduce the effort required to build models. Especially the bad performance of the neural network model seems to confirm this fact. However, while AutoML and stochastic search both have yielded superior performance and AutoML can be recommended based on its optimization potential, there is still potential to increase the effectiveness of hyper-parameter tuning.

Optimization Time

The best performing models we evaluated are based on automatic hyper-parameter optimization:

- **AutoML**, which uses Bayesian optimization to tailor a ensemble of models to the given problem.
- **RFOstochastic**, a random forest regressor with randomly optimized hyper-parameters.

Each of these algorithms was given a time limit for hyper-parameter tuning. In order to see the effect of this time limit, we applied both approaches using different limits. The RMSE scores,

computed as mean over 40 10-fold cross validation scores, of the resulting models are shown in Figure S1

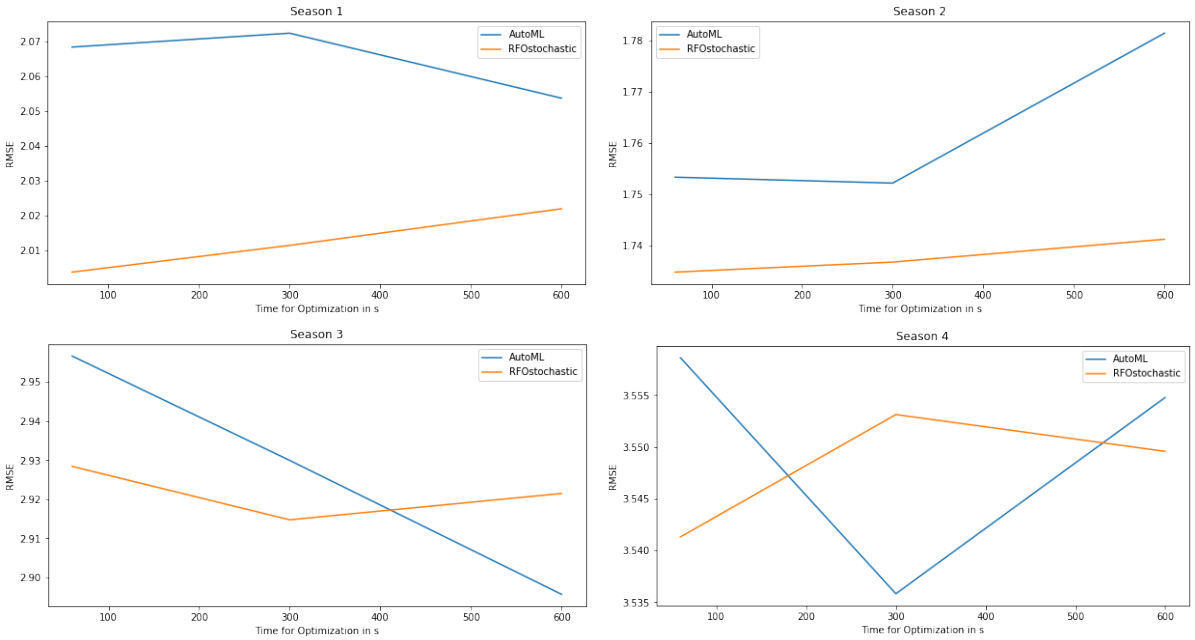


Figure S1: RMSE scores resulting from models (AutoML and RFOstochastic) with different hyper-parameter optimization times. Each of the graphs shows the results for one season. The x-axis of each graph indicates the given optimization times. We evaluated at 60, 300, and 600 seconds. While some effects are visible (e.g., RFOstochastic in Season 1 and 2), overall, the there is no clear trend towards improved performance with increasing optimization time.

First of all, none of those reported performance values differs significantly from the other scores. While some effects are visible, e.g., RFOstochastic seems to perform better with an increasing time limit on Season 1 and 2, there is no clear trend. However, the effect of longer optimization times may improve when going beyond a training time of ten minutes (600s).[S47] However the choice of the optimization time for automated hyper-parameter tuning is an open problem and, thus, is out of the scope of this work.

References

(S1) Briggs, D. J.; Collins, S.; Elliott, P.; Fischer, P.; Kingham, S.; Lebret, E.; Pryl, K.; van Reeuwijk, H.; Smallbone, K.; van der Veen, A. Mapping urban air pollution using GIS:

a regression-based approach. *International Journal of Geographical Information Science* **1997**, *11*, 699–718.

(S2) Stedman, J. R.; Vincent, K. J.; Campbell, G. W.; Goodwin, J. W.; Downing, C. E. New high resolution maps of estimated background ambient NOx and NO2 concentrations in the U.K. *Atmospheric Environment* **1997**, *31*, 3591 – 3602.

(S3) Briggs, D. J.; de Hoogh, C.; Gulliver, J.; Wills, J.; Elliott, P.; Kingham, S.; Smallbone, K. A regression-based method for mapping traffic-related air pollution: application and testing in four contrasting urban environments. *Science of The Total Environment* **2000**, *253*, 151 – 167.

(S4) Carr, D.; von Ehrenstein, O.; Weiland, S.; Wagner, C.; Wellie, O.; Nicolai, T.; von Mutius, E. Modeling Annual Benzene, Toluene, NO2, and Soot Concentrations on the Basis of Road Traffic Characteristics. *Environmental Research* **2002**, *90*, 111 – 118.

(S5) Brauer, M.; Hoek, G.; van Vliet, P.; Meliefste, K.; Fischer, P.; Gehring, U.; Heinrich, J.; Cyrys, J.; Bellander, T.; Lewne, M.; Brunekreef, B. Estimating Long-Term Average Particulate Air Pollution Concentrations: Application of Traffic Indicators and Geographic Information Systems. *Epidemiology* **2003**, *14*, 228–239.

(S6) Gilbert, N. L.; Goldberg, M. S.; Beckerman, B.; Brook, J. R.; Jerrett, M. Assessing Spatial Variability of Ambient Nitrogen Dioxide in Montréal, Canada, with a Land-Use Regression Model. *Journal of the Air & Waste Management Association* **2005**, *55*, 1059–1063.

(S7) Kanaroglou, P. S.; Jerrett, M.; Morrison, J.; Beckerman, B.; Arain, M. A.; Gilbert, N. L.; Brook, J. R. Establishing an air pollution monitoring network for intra-urban population exposure assessment: A location-allocation approach. *Atmospheric Environment* **2005**, *39*, 2399 – 2409, 12th International Symposium, Transport and Air Pollution.

(S8) Gonzales, M.; Qualls, C.; Hudgens, E.; Neas, L. Characterization of a spatial gradient of

nitrogen dioxide across a United States–Mexico border city during winter. *Science of The Total Environment* **2005**, *337*, 163 – 173.

(S9) Smith, L.; Mukerjee, S.; Gonzales, M.; Stallings, C.; Neas, L.; Norris, G.; Özkaynak, H. Use of GIS and ancillary variables to predict volatile organic compound and nitrogen dioxide levels at unmonitored locations. *Atmospheric Environment* **2006**, *40*, 3773 – 3787.

(S10) Rossa, Z.; Englishb, P. B.; Scalfc, R.; Gunierb, R.; Smorodinskyb, S.; Wallb, S.; Jerrettd, M. Nitrogen dioxide prediction in Southern California using land use regression modeling: potential for environmental health analyses. *Journal of Exposure Science and Environmental Epidemiology* **2006**,

(S11) Hochadel, M.; Heinrich, J.; Gehring, U.; Morgenstern, V.; Kuhlbusch, T.; Link, E.; Wichmann, H.-E.; Krämer, U. Predicting long-term average concentrations of traffic-related air pollutants using GIS-based information. *Atmospheric Environment* **2006**, *40*, 542 – 553.

(S12) Sahsuvaroglu, T.; Arain, A.; Kanaroglou, P.; Finkelstein, N.; Newbold, B.; Jerrett, M.; Beckerman, B.; Brook, J.; Finkelstein, M.; Gilbert, N. L. A Land Use Regression Model for Predicting Ambient Concentrations of Nitrogen Dioxide in Hamilton, Ontario, Canada. *Journal of the Air & Waste Management Association* **2006**, *56*, 1059–1069.

(S13) Beelen, R.; Hoek, G.; Fischer, P.; van den Brandt, P. A.; Brunekreef, B. Estimated long-term outdoor air pollution concentrations in a cohort study. *Atmospheric Environment* **2007**, *41*, 1343 – 1358.

(S14) Ross, Z.; Jerrett, M.; Ito, K.; Tempalski, B.; Thurston, G. D. A land use regression for predicting fine particulate matter concentrations in the New York City region. *Atmospheric Environment* **2007**, *41*, 2255 – 2269.

(S15) Ryan, P. H.; LeMasters, G. K.; Biswas, P.; Levin, L.; Hu, S.; Lindsey, M.; Bernstein, D. I.; Lockey, J.; Villareal, M.; Hershey, G. K. K. A comparison of proximity and land use regres-

sion traffic exposure models and wheezing in infants. *Environmental health perspectives* **2007**, *115*, 278.

(S16) Morgenstern, V.; Zutavern, A.; Cyrys, J.; Brockow, I.; Gehring, U.; Koletzko, S.; Bauer, C.-P.; Reinhardt, D.; Wichmann, H.-E.; Heinrich, J. Respiratory health and individual estimated exposure to traffic-related air pollutants in a cohort of young children. *Occupational and environmental medicine* **2007**, *64*, 8–16.

(S17) Moore, D.; Jerrett, M.; Mack, W.; Künzli, N. A land use regression model for predicting ambient fine particulate matter across Los Angeles, CA. *Journal of Environmental Monitoring* **2007**, *9*, 246–252.

(S18) Madsen, C.; Carlsen, K. C. L.; Hoek, G.; Oftedal, B.; Nafstad, P.; Meliefste, K.; Jacobsen, R.; Nystad, W.; Carlsen, K.-H.; Brunekreef, B. Modeling the intra-urban variability of outdoor traffic pollution in Oslo, Norway—A GA2LEN project. *Atmospheric Environment* **2007**, *41*, 7500 – 7511.

(S19) Henderson, S. B.; Beckerman, B.; Jerrett, M.; Brauer, M. Application of Land Use Regression to Estimate Long-Term Concentrations of Traffic-Related Nitrogen Oxides and Fine Particulate Matter. *Environmental Science & Technology* **2007**, *41*, 2422–2428, PMID: 17438795.

(S20) Jerrett, M.; Arain, M. A.; Kanaroglou, P.; Beckerman, B.; Crouse, D.; Gilbert, N. L.; Brook, J. R.; Finkelstein, N.; Finkelstein, M. M. Modeling the Intraurban Variability of Ambient Traffic Pollution in Toronto, Canada. *Journal of Toxicology and Environmental Health, Part A* **2007**, *70*, 200–212.

(S21) Arain, M.; Blair, R.; Finkelstein, N.; Brook, J.; Sahsuvaroglu, T.; Beckerman, B.; Zhang, L.; Jerrett, M. The use of wind fields in a land use regression model to predict air pollution concentrations for health exposure studies. *Atmospheric Environment* **2007**, *41*, 3453 – 3464.

(S22) Rosenlund, M.; Forastiere, F.; Stafoggia, M.; Porta, D.; Perucci, M.; Ranzi, A.; Nussio, F.; Perucci, C. A. Comparison of regression models with land-use and emissions data to predict the spatial distribution of traffic-related air pollution in Rome. *Journal of Exposure Science and Environmental Epidemiology* **2008**, *18*, 192–199.

(S23) Aguilera, I.; Sunyer, J.; Fernández-Patier, R.; Hoek, G.; Aguirre-Alfaro, A.; Meliefste, K.; Bomboi-Mingarro, M. T.; Nieuwenhuijsen, M. J.; Herce-Garraleta, D.; Brunekreef, B. Estimation of outdoor NO x, NO2, and BTEX exposure in a cohort of pregnant women using land use regression modeling. *Environmental science & technology* **2007**, *42*, 815–821.

(S24) Wheeler, A. J.; Smith-Doiron, M.; Xu, X.; Gilbert, N. L.; Brook, J. R. Intra-urban variability of air pollution in Windsor, Ontario—Measurement and modeling for human exposure assessment. *Environmental Research* **2008**, *106*, 7 – 16.

(S25) Gulliver, J.; de Hoogh, K.; Fecht, D.; Vienneau, D.; Briggs, D. Comparative assessment of GIS-based methods and metrics for estimating long-term exposures to air pollution. *Atmospheric Environment* **2011**, *45*, 7072 – 7080.

(S26) Su, J.; Jerrett, M.; Beckerman, B. A distance-decay variable selection strategy for land use regression modeling of ambient air pollution exposures. *Science of The Total Environment* **2009**, *407*, 3890 – 3898, Thematic Issue - BioMicroWorld Conference.

(S27) Rivera, M.; Basagaña, X.; Aguilera, I.; Agis, D.; Bouso, L.; Foraster, M.; Medina-Ramón, M.; Pey, J.; Künzli, N.; Hoek, G. Spatial distribution of ultrafine particles in urban settings: A land use regression model. *Atmospheric Environment* **2012**, *54*, 657 – 666.

(S28) Abernethy, R. C.; Allen, R. W.; McKendry, I. G.; Brauer, M. A Land Use Regression Model for Ultrafine Particles in Vancouver, Canada. *Environmental Science & Technology* **2013**, *47*, 5217–5225, PMID: 23550900.

(S29) Saraswat, A.; Apte, J. S.; Kandlikar, M.; Brauer, M.; Henderson, S. B.; Marshall, J. D. Spatiotemporal Land Use Regression Models of Fine, Ultrafine, and Black Carbon Particulate

Matter in New Delhi, India. *Environmental Science & Technology* **2013**, *47*, 12903–12911, PMID: 24087939.

(S30) Dons, E.; Poppel, M. V.; Kochan, B.; Wets, G.; Panis, L. I. Modeling temporal and spatial variability of traffic-related air pollution: Hourly land use regression models for black carbon. *Atmospheric Environment* **2013**, *74*, 237 – 246.

(S31) Ragettli, M. S.; Ducret-Stich, R. E.; Foraster, M.; Morelli, X.; Aguilera, I.; Basagaña, X.; Corradi, E.; Ineichen, A.; Tsai, M.-Y.; Probst-Hensch, N.; Rivera, M.; Slama, R.; Künzli, N.; Phuleria, H. C. Spatio-temporal variation of urban ultrafine particle number concentrations. *Atmospheric Environment* **2014**, *96*, 275 – 283.

(S32) Montagne, D. R.; Hoek, G.; Klompmaker, J. O.; Wang, M.; Meliefste, K.; Brunekreef, B. Land Use Regression Models for Ultrafine Particles and Black Carbon Based on Short-Term Monitoring Predict Past Spatial Variation. *Environmental Science & Technology* **2015**, *49*, 8712–8720, PMID: 26079151.

(S33) Kerckhoffs, J.; Hoek, G.; Messier, K. P.; Brunekreef, B.; Meliefste, K.; Klompmaker, J. O.; Vermeulen, R. Comparison of Ultrafine Particle and Black Carbon Concentration Predictions from a Mobile and Short-Term Stationary Land-Use Regression Model. *Environmental Science & Technology* **2016**, *50*, 12894–12902, PMID: 27809494.

(S34) Zwack, L. M.; Paciorek, C. J.; Spengler, J. D.; Levy, J. I. Characterizing local traffic contributions to particulate air pollution in street canyons using mobile monitoring techniques. *Atmospheric Environment* **2011**, *45*, 2507 – 2514.

(S35) Patton, A. P.; Collins, C.; Naumova, E. N.; Zamore, W.; Brugge, D.; Durant, J. L. An Hourly Regression Model for Ultrafine Particles in a Near-Highway Urban Area. *Environmental Science & Technology* **2014**, *48*, 3272–3280, PMID: 24559198.

(S36) Larson, T.; Henderson, S. B.; Brauer, M. Mobile Monitoring of Particle Light Absorption

Coefficient in an Urban Area as a Basis for Land Use Regression. *Environmental Science & Technology* **2009**, *43*, 4672–4678, PMID: 19673250.

(S37) Sîrbu, A.; Becker, M.; Caminiti, S.; De Baets, B.; Elen, B.; Francis, L.; Gravino, P.; Hotho, A.; Ingarra, S.; Loreto, V.; Molino, A.; Mueller, J.; Peters, J.; Ricchiuti, F.; Saracino, F.; Servedio, V. D. P.; Stumme, G.; Theunis, J.; Tria, F.; Van den Bossche, J. Participatory Patterns in an International Air Quality Monitoring Initiative. *PLOS ONE* **2015**, *10*, 1–19.

(S38) Hasenfratz, D.; Saukh, O.; Walser, C.; Hueglin, C.; Fierz, M.; Thiele, L. Pushing the spatio-temporal resolution limit of urban air pollution maps. 2014 IEEE International Conference on Pervasive Computing and Communications (PerCom). 2014; pp 69–77.

(S39) Hankey, S.; Marshall, J. D. Land Use Regression Models of On-Road Particulate Air Pollution (Particle Number, Black Carbon, PM2.5, Particle Size) Using Mobile Monitoring. *Environmental Science & Technology* **2015**, *49*, 9194–9202, PMID: 26134458.

(S40) Shi, J. P.; Harrison, R. M. Regression modelling of hourly NOx and NO2 concentrations in urban air in London. *Atmospheric Environment* **1997**, *31*, 4081 – 4094.

(S41) Corani, G. Air quality prediction in Milan: feed-forward neural networks, pruned neural networks and lazy learning. *Ecological Modelling* **2005**, *185*, 513 – 529.

(S42) Marjovi, A.; Arfire, A.; Martinoli, A. High Resolution Air Pollution Maps in Urban Environments Using Mobile Sensor Networks. 2015 International Conference on Distributed Computing in Sensor Systems. 2015; pp 11–20.

(S43) Fuller, R.; Groom, G.; Jones, A. Land cover map of Great Britain. An automated classification of Landsat Thematic Mapper data. *Photogrammetric Engineering and Remote Sensing* **1994**, *60*.

(S44) Hoek, G.; Meliefste, K.; Brauer, M.; van Vliet, P.; Brunekreef, B.; Fischer, P.; Lebret, E.; Cyrys, J.; Gehring, U.; Heinrich, A. Risk assessment of exposure to traffic-related air pollution for the development of inhalant allergy, asthma and other chronic respiratory conditions in children (TRAPCA). *Final Report. Utrecht: IRAS University* **2001**,

(S45) Bellemans, T.; Kochan, B.; Janssens, D.; Wets, G.; Arentze, T.; Timmermans, H. Implementation framework and development trajectory of FEATHERS activity-based simulation platform. *Transportation Research Record: Journal of the Transportation Research Board* **2010**, 111–119.

(S46) Hastie, T.; Tibshirani, R. *Generalized additive models*; Wiley Online Library, 1990.

(S47) Blum, M.; Feurer, M.; Klein, A.; Springenberg, J.; Hutter, F.; Eggensperger, K. Efficient and Robust Automated Machine Learning. 2015; `http://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning`.

(S48) Hu, X.; Waller, L. A.; Al-Hamdan, M. Z.; Crosson, W. L.; Estes Jr, M. G.; Estes, S. M.; Quattrochi, D. A.; Sarnat, J. A.; Liu, Y. Estimating ground-level PM2. 5 concentrations in the southeastern US using geographically weighted regression. *Environmental Research* **2013**, *121*, 1–10.

Semi-Supervised Learning for Grain Size Distribution Interpolation

Konstantin Kobs(✉), Christian Schäfer, Michael Steininger, Anna Krause, Roland Baumhauer, Heiko Paeth, and Andreas Hotho

University of Würzburg, Würzburg, Germany
{kobs,steininger,anna.krause,hotho}@informatik.uni-wuerzburg.de,
{christian.d.schaefer,baumhauer,heiko.paeth}@uni-wuerzburg.de

Abstract. High-resolution grain size distribution maps for geographical regions are used to model soil-hydrological processes that can be used in climate models. However, measurements are expensive or impossible, which is why interpolation methods are used to fill the gaps between known samples. Common interpolation methods can handle such tasks with few data points since they make strong modeling assumptions regarding soil properties and environmental factors. Neural networks potentially achieve better results as they do not rely on these assumptions and approximate non-linear relationships from data. However, their performance is often severely limited for tasks like grain size distribution interpolation due to their requirement for many training examples. Semi-supervised learning may improve their performance on this task by taking widely available unlabeled auxiliary data (e.g. altitude) into account.

We propose a novel semi-supervised training strategy for spatial interpolation tasks that pre-trains a neural network on weak labels obtained by methods with stronger assumptions and then fine-tunes the network on the small labeled dataset. In our research area, our proposed strategy improves the performance of a supervised neural network and outperforms other commonly used interpolation methods.

Keywords: Spatial interpolation · Semi-supervised learning · Neural networks

1 Introduction

The composition of different grain sizes in the soil affects many hydrological processes such as groundwater recharge, infiltration rates or surface flow. For example, soils with dominating clay fractions (grain size $\leq 0.002\,\text{mm}$) retain water better than sandy soils ($0.063\,\text{mm} <$ grain size $\leq 2.0\,\text{mm}$). Given accurate grain size distribution maps, it is possible to estimate hydrological parameters for environmental modelling purposes, e.g. regional climate models. Since sampling is expensive or even impossible due to inaccessible terrain, spatial interpolation methods are used to estimate grain size distributions for unknown locations.

A. Del Bimbo et al. (Eds.): ICPR 2020 Workshops, LNCS 12666, pp. 34–44, 2021.
https://doi.org/10.1007/978-3-030-68780-9_4

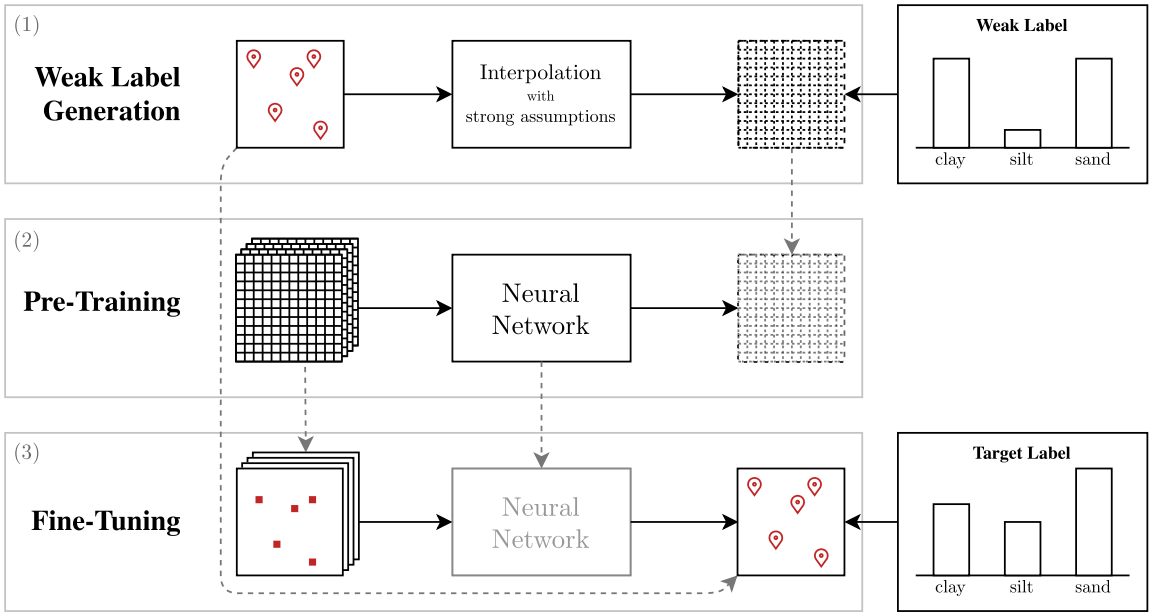


Fig. 1. In our proposed semi-supervised training method, (1) a spatial interpolation method with strong assumptions is trained on the labeled dataset. (2) The neural network is pre-trained on weak labels obtained by applying the interpolation method to the unlabeled data. The network gets locations and auxiliary data as inputs. (3) It is then fine-tuned on the labeled dataset.

A model for grain size distribution interpolation has the following requirements: (1) The model input is a location with (potentially) additional auxiliary data (e.g. altitude). (2) The model outputs distributions across the grain size classes (clay, silt, sand) for each unknown location. (3) The model works with few labeled data points, since soil samples are rare.

Distance based interpolation methods such as k Nearest Neighbors or Inverse Distance Weighting can output distributions and are applicable to small labeled datasets due to their strong assumptions. However, they do not take auxiliary data into account which can benefit performance [11,17]. Neural networks can learn non-linear relationships from data, are able to incorporate additional auxiliary inputs, and are able to output distributions across grain size classes. However, they usually need many labeled training data points [15]. The idea of semi-supervised learning utilizes large unlabeled datasets to support network training [8]. In recent years, most methods for semi-supervised learning were designed for image classification, which are not applicable to our setting.

Therefore, in this paper, we bring semi-supervised learning specifically to the task of grain size distribution interpolation for spatial inputs. We propose a training strategy that makes use of weak labels produced by an interpolation method with stronger modeling assumptions. Figure 1 gives a schematic overview of our proposed three-step process. In our experiments for the region of Lower Franconia, we show that our approach improves the performance of a supervised neural network and outperforms other common interpolation methods. Furthermore, we analyze the effects of the proposed training strategy on model performance.

Our contributions are: (1) We describe a semi-supervised training strategy for neural networks in the spatial domain to interpolate grain size distributions. (2) We compare our strategy to supervised training and common interpolation approaches and show that it outperforms them in our research area. (3) We analyze the resulting model to understand what factors are important for its performance.

2 Related Work

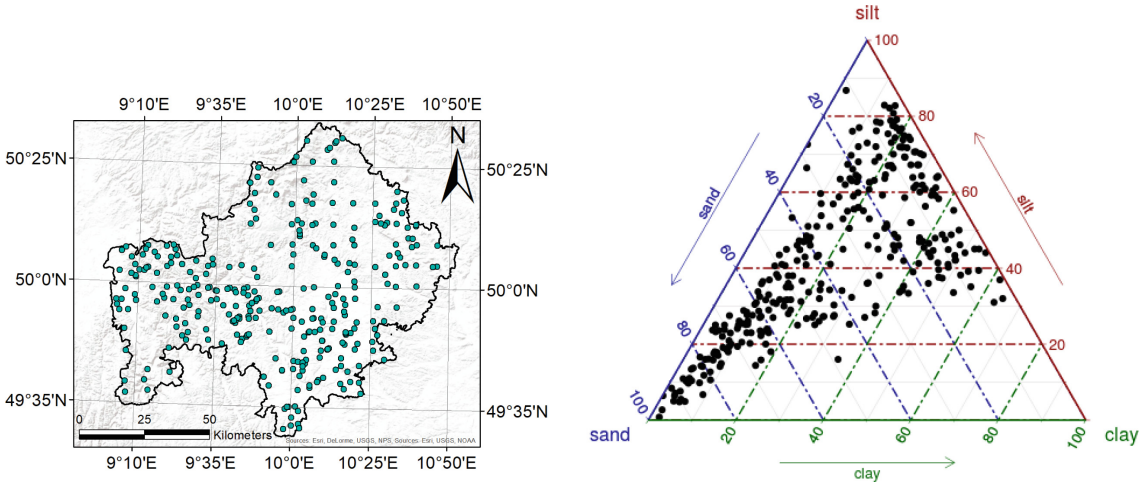
There are various spatial interpolation techniques with different properties used in environmental sciences, e.g. k Nearest Neighbors, Inverse Distance Weighting, or Kriging [16]. Neural networks have been successfully applied in such tasks since they allow auxiliary data as input features and can model non-linear relationships [5,20,23]. However, to obtain robust performance, they need many labeled data points not available in most spatial interpolation tasks [15]. Semi-supervised training promotes the use of large unlabeled datasets to support the training of neural networks with few labeled data points [8]. For image classification, which is the most popular semi-supervised learning task, domain-specific strategies such as image augmentation have been proposed, which are not trivial to apply in our setting. Classification specific approaches such as using the softmax output of the network as confidence for a weak label [26] are not directly applicable to our task, since our desired output is a distribution and not a class.

For our semi-supervised training strategy, we adapt so-called "distant supervision" from other domains [10,14] by training the network on weak labels. Obtaining weak labels from more traditional interpolation methods and fine-tuning the network on labeled data afterwards is a new approach in this area.

3 Research Area and Dataset

In this section, we describe the research area and the dataset we use for the interpolation task. Inputs to the interpolation models are the *latitude*, *longitude*, and multiple features from different auxiliary data sources that we suspect to have an influence on or are influenced by the grain size distribution. While only 315 locations have a target grain size distribution, the auxiliary data is widely available in a fine grid of 25 m × 25 m cells (overall 11 952 963 grid cells).

The research area is Lower Franconia, northern Bavaria, Germany. It covers 8530 km and falls within 49.482°N to 50.566°N and 8.978°E to 10.881°E. The topography of this region is characterized by alluvial zones with surrounding low mountain ranging from 96 m to 927 m in altitude.



(a) Labeled data point locations. Map tiles by ESRI, USGS, NOAA, data by BEA.

(b) Distribution of target grain size distributions.

Fig. 2. Map showing labeled locations and distribution of the labels.

3.1 Target Variable: Grain Size Distribution

Soils are compositions of grain sizes. To get soil conditions for the research area, we use a soil profile database of the Bavarian Environment Agency (BEA)[1]. The database covers detailed information on in-depth grain size distribution on 431 sites in Lower Franconia. The sampling took place in-between 1989 and 2017 and exposes grain size distributions of the fine earth fraction per soil-horizon through combined sieve and pipette analysis [12]. The method of sampling varies between drill cores and complete profile excavations.

While each observed location lists multiple layers, we limit the interpolation task to two dimensions by only using soil information from 14 cm–15 cm as most recorded layers span across this range. This common approach [6] results in 315 labeled locations, shown in Fig. 2a.

Given the detailed grain sizes, we represent each location as a composition of three grain size classes [1]: **clay** (grain size $\leq 0.002\,\text{mm}$), **silt** ($0.002\,\text{mm} <$ grain size $\leq 0.063\,\text{mm}$), and **sand** ($0.063\,\text{mm} <$ grain size $\leq 2.0\,\text{mm}$). Each label is a three dimensional distribution vector, e.g. 20 % clay, 50 % silt, and 30 % sand. The label distribution is shown in Fig. 2b. The task is to estimate this distribution for a location given other locations and auxiliary data.

3.2 Auxiliary Data

While there are only 315 labeled data points, auxiliary data is available for all locations in Lower Franconia (11 952 963 grid cells). For this work, we use a Digital Elevation Model (DEM) and meteorological data to generate ten features for each grid cell: *latitude*, *longitude*, *altitude*, *slope*, *Multi-Scale Topographic*

[1] Unpublished data; reference: https://www.lfu.bayern.de/umweltdaten/.

Position Index (minimum, mean, and maximum), *Topographic Wetness Index*, *temperature*, and *precipitation*, that are explained in the following.

The used DEM provided by the BEA[2] reflects the *altitude* of the terrain surface, excluding buildings and vegetation, resampled to our grid's spatial resolution of 25 m. We derive five additional features through topographic, morphometric and hydrographic analysis [25].

Slope. In basic terrain analysis, *slope* represents the change in elevation over a given distance. For a cell with altitude alt, we calculate the mean altitude over the neighboring cells in north and south direction $\overline{\text{alt}}_{\text{NS}}$ and in west and east direction $\overline{\text{alt}}_{\text{WE}}$. The slope ranges from 0° (a horizontal plane) to 90° and is calculated using $\text{slope} = \frac{180}{\pi \cdot \sqrt{\left(\overline{\text{alt}}_{\text{NS}} - \text{alt}\right)^2 + \left(\overline{\text{alt}}_{\text{WE}} - \text{alt}\right)^2}}$.

Multi-scale Topographic Position Index. The Topographic Position Index (TPI) [24] is defined as the altitude difference between a location of interest and the mean altitude of a square area around it, giving values that indicate local ridges and valleys. We obtain TPIs on multiple scales by altering the side length of the square from 3 grid cells (75 m) to 41 grid cells (1025 m) in steps of two cells, having the current location in the square's center. From the resulting 19 TPIs, we take the *minimum*, *mean*, and *maximum* as features. They describe the morphology of our study area at different scales as numeric factors.

Topographic Wetness Index. To represent spatial variations of soil moisture content and soil water drainage, a *terrain-based wetness index (TWI)* is computed [4]. The index is high for locations where water normally collects due to the topographic setting. It is calculated as a tangent function of the cell's slope angle w.r.t. the cell's area (625 m^2): $\text{TWI} = \ln\left(\frac{625}{\tan(\text{slope})}\right)$.

Meteorological Data. In addition to terrain based features described above, we also obtain meteorological data provided by the German Meteorological Service (DWD). The data reflects the 30-year (1971–2000) means of the monthly averaged mean daily air *temperature* 2 m above the ground and *precipitation*.[3] The grid-based data was obtained by accurate interpolation methods for temperature and precipitation at a resolution of 1 km^2 [19] and resampled to the target grid size of 25 m using nearest neighbor interpolation.

4 Methodology

Given the data described above, we now have a large dataset of unlabeled data as well as a small labeled dataset. A neural network should now learn to estimate the grain size distribution of a location based on the ten input features. To make use of the large unlabeled dataset, we propose a three step semi-supervised training strategy that pre-trains the neural network on weak labels created by an interpolation method with stronger assumptions:

[2] https://geodatenonline.bayern.de/geodatenonline/seiten/dgm_info.

[3] https://opendata.dwd.de/climate_environment/CDC/grids_germany/multi_annual/*air_temperature_mean* and *precipitation*.

1. Weak Label Generation. We apply a common interpolation method such as Inverse Distance Weighting (IDW) on the small labeled dataset. Note that these methods usually do not take auxiliary data into account. Due to the strong modeling assumptions of such algorithms, they are able to work with small datasets. The trained model then estimates the target labels for the large unlabeled dataset, which are used as weak labels in the next step.

2. Pre-training. The neural network is pre-trained using the large amount of available weakly labeled data, thus being exposed to the property assumptions of the weak label generator. This way, the network learns representations from all input features, including the auxiliary data, and is guided to create more realistic outputs. Since interpolation methods such as IDW represent the location information as distances, the network has to learn from different features, as we will show in Sect. 6.1. Calculating the euclidean distance from locations is hard for the network, therefore it tries to find other correlations as well.

3. Fine-tuning. The pre-trained network is fine-tuned on the labeled dataset. This reinforces or weakens some correlations the network has found. For fine-tuning, a smaller learning rate is used in order to keep the previously trained weights intact. The resulting model can then be used on all locations.

5 Experiments

Now, we compare our self-supervised training strategy to the traditional supervised method and other common interpolation methods on the grain size distribution task. Note that not all methods can output distributions, so we will only apply methods that are able to handle this task-specific output type.

5.1 Methods

Mean. Always predicts the mean of all training examples. As the average of multiple distributions is also a distribution, the prediction is valid.

k Nearest Neighbors (kNN). Calculates the average label of the nearest k training locations [2]. We set $k = 3$ based on a parameter search on validation data for $k \in \{1, \ldots, 10\}$.

Inverse Distance Weighting (IDW). Same as kNN, but the average is inversely weighted based on the distance to a labeled location [22]. A parameter search for $k \in \{1, \ldots, 10\}$ results in $k = 7$.

Multilayer Perceptron (MLP). Trains a Multilayer Perceptron on the labeled dataset in a supervised learning setting. The ten-dimensional input is normalized to zero mean and unit variance. It is then fed through three hidden layers with 256 neurons each with ReLU activation functions [9] in a batch of size 1024. The three-dimensional output is then converted to a probability distribution by applying the softmax activation function. These hyperparameters have been

Table 1. Test results (mean ± standard deviation) for each model. Best values are written in bold.

	MAE	MSE	JSD
Mean	0.5210 ± 0:0384	0.1337 ± 0.0183	0.0549 ± 0.0076
kNN	0.4267 ± 0.0412	0.1011 ± 0.0223	0.0398 ± 0.0090
IDW	0.4188 ± 0.0417	0.0954 ± 0.0225	0.0381 ± 0.0090
MLP	0.4361 ± 0.0552	0.1068 ± 0.0251	0.0426 ± 0.0088
SemiMLP (after pre-training)	0.4781 ± 0.0577	0.1296 ± 0.0283	0.0497 ± 0.0099
SemiMLP (after fine-tuning)	**0.4078 ± 0.0445**	**0.0952 ± 0.0195**	**0.0377 ± 0.0077**

found on validation data. The standard cross entropy loss function is used that allows distributions as targets. The network is optimized with Adam [13] and a learning rate of 10^{-1} for at most 1000 epochs. Early stopping [18] stops the training if the validation loss does not improve at least 10^{-5} for ten epochs.

Semi-supervised MLP (SemiMLP). We apply our semi-supervised training strategy to the same MLP architecture as above. We generate weak labels using the IDW baseline with $k = 7$ as it achieved the best baseline validation results. We train the network with learning rates 10^{-1} and 10^{-3} for pre-training and fine-tuning, respectively.

5.2 Evaluation

To evaluate the methods described above, we perform a ten-fold cross-validation (i.e. 31 or 32 examples per fold) using the labeled dataset. We average over 50 repetitions to account for the random initialization of the neural networks. Three metrics are used for evaluation: **Mean Absolute Error (MAE)**, **Mean Squared Error (MSE)**, and **Jensen-Shannon Divergence (JSD)**. While MAE and MSE compute the mean (absolute and squared) deviation from the correct values, JSD is specifically designed to measure the difference between two distributions [7]. Note that MAE and MSE sum the errors up for an example before averaging over all examples.

6 Results

Table 1 shows the test results for all models. The model with our training strategy (SemiMLP) yields the best test results. While the supervised MLP performs worse than kNN, the fine-tuned SemiMLP even improves the performance of the IDW baseline. In fact, a Wilcoxon signed rank test ($\alpha = 0.01$) on the MSE indicates that the improvement w.r.t. IDW is significant. We suspect that the network's improvement comes from having direct access to locations as well as auxiliary data that it uses during training, while IDW only relies on distances between locations as inputs.

6.1 Analysis

Pre-training Matters. For our experiments, we altered the MLP baseline by adding the pre-training step to obtain SemiMLP, while the architecture and preprocessing were fixed. Thus, SemiMLP's better performance compared to MLP (cf. Table 1) shows that pre-training has a positive effect on SemiMLP. Pre-training the network seems to build better representations for the downstream task than random initialization.

Fine-tuning Matters. While it helps, pre-training alone does not give superior performance. Table 1 shows that only pre-training on weak labels gives worse performance than most baselines and the supervised MLP. This indicates that the network is not able to imitate the IDW baseline, which generated the weak labels. This may be due to IDW using distances between new and labeled locations to assess its predictions. SemiMLP does not get distance information as input and is not able to directly access the labeled dataset. Thus, it learns a surrogate function that fits the training data but will not exactly match IDW's output for new data points. Also, SemiMLP gets more features than IDW, increasing the chance that the network exploits other correlations to predict the output. After the fine-tuning step, the method is superior to all baselines.

Auxiliary Data Matters. The features that may be influenced by or influence the target variable also have an effect on the performance. To investigate this, we apply the permutation importance for feature evaluation method [3] that permutes the values of a feature to see how much the predictive quality of the trained model changes. The more important a feature is, the higher the drop in performance if its input is altered. We average the features' importances for each test fold over ten different permutations to get more robust results.

Figure 3 shows the resulting feature importances. Besides location, the features temperature, precipitation, and altitude have the largest influence. According to previous research, soil is formed by the alteration of present bedrock under the influence of *climate*, *relief*, *organisms*, and *human activity* over time [21]. Since we do not provide features describing organisms and human activity, the model focuses on climatic (30-year means of temperature and precipitation) and relief-based (altitude) influences. While we expected other relief-based features such as TPI or TWI to be more important for the model, altitude and location seem to be descriptive enough.

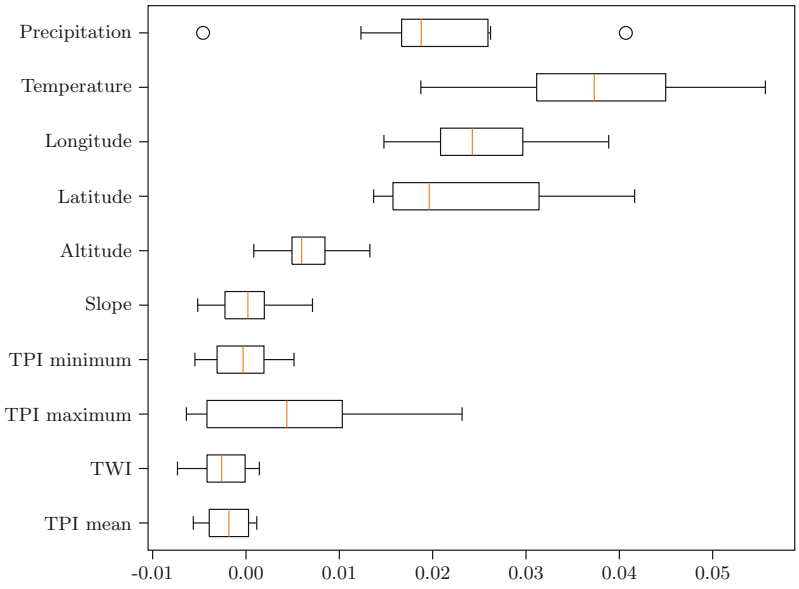


Fig. 3. Drop in MAE performance when the feature column was permuted.

7 Discussion

Neural networks make no modeling assumptions for the interpolation task. Compared to common interpolation methods, the network can model non-linear relationships in the data and can utilize any kind of auxiliary data. Our method circumvents the necessity of large training datasets by guiding the network towards more realistic outputs using weak labels before fine-tuning on few real labels. It is very easy to replace the weak label generator with a potentially better interpolation method. The required pre-training of the network on weakly labeled data takes extensively longer. However, depending on the neural network architecture, input data, and size of the research area, inference can be faster than other approaches, as we can compute outputs in batches on specialized hardware without any distance calculations.

As stated in Sect. 3, we restrict this work to the two-dimensional case of grain size distribution interpolation. While depth information is expected to increase performance, it is not trivial to use it in the weak label generation methods. Labeled locations usually have large distances (hundreds to thousands of meters), while labeled soil layers have very small distances (millimeters to few centimeters). Distance based approaches such as IDW will only take the nearest labeled location into account and average its soil layers as these are overall the closest to the desired location. While this is not resolved, building a model for each depth layer is the simplest approach that we can apply in practice.

8 Conclusion

In this paper we have proposed a semi-supervised training method for spatial interpolation tasks. For our grain size distribution task, additional pre-training on weak labels improved the network's performance compared to supervised learning and common interpolation methods. Testing other weak label generators and sampling strategies to optimize pre-training remains future work. Mixing weak labels from methods with different modeling assumptions might enrich the learned representations of the network. Future challenges include adding the depth dimension, allowing the exploitation of soil layer relations. Further, we will evaluate the interpolated map in a soil-hydrological simulation model.

Acknowledgements. This research was conducted in the BigData@Geo project supported by the European Regional Development Fund (ERDF).

References

1. Ad-hoc-AG Boden: Bodenkundliche Kartieranleitung. Schweizerbart, 5 edn. (2005)
2. Altman, N.S.: An Introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. **46**(3) (1992)
3. Breiman, L.: Random forests. Mach. Learn. **45**(1) (2001)
4. Böhner, J., Selige, T.: Spatial prediction of soil attributes using terrain analysis and climate regionalization. Gottinger Geographische Abhandlungen **115** (2002)
5. Dai, F., Zhou, Q., Lv, Z., Wang, X., Liu, G.: Spatial prediction of soil organic matter content integrating artificial neural network and ordinary kriging in Tibetan Plateau. Ecol. Indicators **45** (2014)
6. Deshmukh, K.K., Aher, S.P.: Particle Size Analysis of Soils and Its Interpolation using GIS Technique from Sangamner Area, vol. 3. Maharashtra, India (2014)
7. Endres, D.M., Schindelin, J.E.: A new metric for probability distributions. IEEE Trans-IT **49**(7) (2003)
8. van Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. Mach. Learn. **109**(2), 373–440 (2019). https://doi.org/10.1007/s10994-019-05855-6
9. Glorot, X., Bordes, A., Bengio, Y.: Deep Sparse rectifier neural networks. In: 14th AISTATS (2011)
10. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N project report, Stanford, vol. 1(12) (2009)
11. Hengl, T.: A Practical Guide to Geostatistical Mapping. 2. extended edn. (2009)
12. ISO Central Secretary: Soil quality - determination of particle size distribution in mineral soil material - method by sieving and sedimentation. Technical Report (2009)
13. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv:1412.6980 (2017)
14. Kobs, K., et al.: Emote-controlled: obtaining implicit viewer feedback through emote-based sentiment analysis on comments of popular twitch.tv channels. TSC **3**(2) (2020)
15. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553) (2015)
16. Li, J., Heap, A.D.: Spatial interpolation methods applied in the environmental sciences: a review. Environ. Model. Softw. **53** (2014)

17. Meyer, S.: Climate change impact assessment under data scarcity. Dissertation, LMU München (2016)
18. Orr, G.B., Müller, K.-R. (eds.): Neural Networks: Tricks of the Trade. LNCS, vol. 1524. Springer, Heidelberg (1998). https://doi.org/10.1007/3-540-49430-8
19. Rauthe, M., Steiner, H., Riediger, U., Mazurkiewicz, A., Gratzki, A.: A central European precipitation climatology part i: Generation and validation of a high-resolution gridded daily data set (hyras). Meteorologische Zeitschrift **22**(3) (2013)
20. Rezaei, K., et al.: Feed forward neural network and interpolation function models to predict the soil and subsurface sediments distribution in Bam. Iran. Acta Geophysica **57**(2) (2009)
21. Semmel, A.: Relief, Gestein. Boden. Wiss, Buchges (1991)
22. Shepard, D.: A two-dimensional interpolation function for irregularly-spaced data. In: 23rd ACM National Conference (1968)
23. Tarasov, D., Buevich, A., Sergeev, A., Shichkin, A.: High variation topsoil pollution forecasting in the Russian Subarctic: Using artificial neural networks combined with residual kriging. Appl. Geochem. **88** (2018)
24. Weiss, A.: Topographic position and landforms analysis. In: Poster presentation, ESRI user Conference, vol. 200, San Diego, CA
25. Wilson, J.P.: Terrain analysis. Wiley (2000)
26. Xie, Q., Luong, M.T., Hovy, E., Le, Q.V.: Self-training with noisy student improves imagenet classification. In: IEEE/CVF CVPR (2020)

C. Declaration of Own Contributions

1. Michael Steininger, Konstantin Kobs, Albin Zehe, Florian Lautenschlager, Martin Becker, and Andreas Hotho. "MapLUR: Exploring a New Paradigm for Estimating Air Pollution Using Deep Learning on Map Images". In: *ACM Trans. Spatial Algorithms Syst.* 6.3 (2020)

   M.S., F.L., and M.B. conceived the methodology. M.S. designed and implemented the MapLUR model, conceived the experiments as well as the analysis using guided backpropagation and carried them out. K.K. contributed the analysis using artificial images and wrote the accompanying section in the publication. F.L. generated features for conventional ML models. M.S. designed visualizations with contributions from K.K. for the publication. M.S., K.K., A.Z., F.L., M.B., and A.H. analyzed and discussed the results. M.S. wrote the publication with substantial contributions from A.Z. and M.B. as well as contributions from K.K., F.L., and A.H.

   The task of estimating pollution based on map or satellite images was already proposed in an experiment in M.S.'s master thesis "Learning the relation of pollution to land usage" but both works have little in common otherwise. Specifically, the paper formulates a novel data-driven paradigm for land-use regression (which is not in the master thesis), proposes a different model, conducts experiments on a different dataset, deals with the estimation of a different pollutant (NO2 instead of UFP), shows good results (the model in the master thesis was not able to learn the task successfully), includes a study on the influence of input area size on model performance, and includes an in-depth analysis on what parts of the input the model pays attention to.

2. Andrzej Dulny, Michael Steininger, Florian Lautenschlager, Anna Krause, and Andreas Hotho. "Evaluating the multi-task learning approach for land use regression modelling of air pollution". In: *Journal of Physics: Conference Series* 1834.1 (2021), p. 012004

   M.S. conceived the idea. M.S. and A.D. conceived the methodology and experiments. A.D. implemented the method and carried the experiments out. A.D., M.S., F.L., A.K., and A.H. analyzed and discussed the results. A.D. wrote the publication with contributions from M.S., F.L., A.K., and A.H.

   This work is based on A.D.'s master thesis, in which large parts of this work are also included. M.S. was the advisor on the master thesis and contributed to both works the idea and core parts of the methodology as well as the experimental setup, the latter two of which were then refined in cooperation with A.D. M.S. further

helped analyzing results during the creation of A.D.'s thesis and thereafter for the paper. As the advisor for the thesis, M.S. guided the process of writing the paper which changed the text significantly.

3. Michael Steininger, Daniel Abel, Katrin Ziegler, Anna Krause, Heiko Paeth, and Andreas Hotho. "Deep Learning for Climate Model Output Statistics". In: *NeurIPS 2020 Workshop on Tackling Climate Change with Machine Learning* (2020)

   M.S. conceived the methodology and implemented the ConvMOS architecture. M.S. conceived the experiments and carried them out. D.A. and K.Z. provided data for the climate model, elevation and weather observations. M.S., D.A., K.Z., A.K., H.P., and A.H. analyzed and discussed the results. M.S. wrote the publication with D.A. and K.Z. writing a description of the data used and other contributions from A.K., H.P., and A.H.

4. Michael Steininger, Daniel Abel, Katrin Ziegler, Anna Krause, Heiko Paeth, and Andreas Hotho. "ConvMOS: Climate Model Output Statistics with Deep Learning". In: *Data Mining and Knowledge Discovery* (2022)

   M.S. conceived the methodology and implemented the ConvMOS architecture. M.S. conceived the experiments and carried them out. D.A. and K.Z. provided data for the climate model, elevation and weather observations. M.S., D.A., K.Z., A.K., H.P., and A.H. analyzed and discussed the results. M.S. wrote the publication with D.A. and K.Z. writing a description of the data used and other contributions from A.K., H.P., and A.H.

5. Michael Steininger, Konstantin Kobs, Padraig Davidson, Anna Krause, and Andreas Hotho. "Density-based weighting for imbalanced regression". In: *Machine Learning* 110.8 (2021), pp. 2187–2211

   M.S. conceived the methodology and implemented DenseWeight and DenseLoss. M.S. conceived the experiments and carried them out. M.S. designed visualizations with contributions from K.K. for the publication. M.S., K.K., P.D., A.K., and A.H. analyzed and discussed the results. M.S. wrote the publication with contributions from K.K., P.D., A.K., and A.H.

6. Florian Lautenschlager, Martin Becker, Konstantin Kobs, Michael Steininger, Padraig Davidson, Anna Krause, and Andreas Hotho. "OpenLUR: Off-the-shelf air pollution modeling with open features and machine learning". In: *Atmospheric Environment* 233 (2020), p. 117535

   F.L., M.B., and A.H. conceptualized the work. F.L. and M.B. conceived the methodology. F.L. implemented the software and applied formal techniques to analyze the study data. F.L., M.B., K.K., M.S., P.D., A.K., and A.H. analyzed and discussed the results. F.L. wrote the publication with contributions from M.B., K.K., M.S., P.D., A.K., and A.H.

7. Konstantin Kobs, Christian Schäfer, Michael Steininger, Anna Krause, Roland Baumhauer, Heiko Paeth, and Andreas Hotho. "Semi-Supervised Learning for Grain

Size Distribution Interpolation". In: *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part VI*. 2021, pp. 34–44

K.K. conceived the methodology, implemented the method, and conceived the experiments. K.K. carried the experiments out with support from M.S. for adapting the code to work in a Kubernetes cluster. C.S. contributed the soil data. K.K., C.S., and M.S. designed and implemented visualizations. K.K., C.S., M.S., A.K., R.B., H.P., and A.H. analyzed and discussed the results. K.K. and C.S. wrote the publication with contributions from M.S., A.K., R.B., H.P., and A.H.

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015.

[2] Ursula Ackermann-Liebrich, Philippe Leuenberger, Joel Schwartz, Christian Schindler, Christian Monn, G Bolognini, JP Bongard, O Brändli, G Domenighetti, S Elsasser, et al. "Lung function and long term exposure to air pollutants in Switzerland. Study on Air Pollution and Lung Diseases in Adults (SAPALDIA) Team." In: *American journal of respiratory and critical care medicine* 155.1 (1997), pp. 122–129.

[3] Matthew Adams. "Advancing the use of mobile monitoring data for air pollution modelling". PhD thesis. McMaster University, 2015.

[4] Madhoolika Agrawal. "Effects of air pollution on agriculture: an issue of national concern". In: *Natl Acad Sci Lett* 28.3/4 (2005), pp. 93–106.

[5] Kamal Ahmed, Shamsuddin Shahid, Nadeem Nawaz, and Najeebullah Khan. "Modeling climate change impacts on precipitation in arid regions of Pakistan: a non-local model output statistics downscaling approach". In: *Theoretical and Applied Climatology* 137.1-2 (2019), pp. 1347–1364.

[6] Air Quality Team (Greater London Authority). *London Atmospheric Emissions Inventory 2013*. 2016.

[7] Ata Akbari Asanjan, Tiantian Yang, Kuolin Hsu, Soroosh Sorooshian, Junqiang Lin, and Qidong Peng. "Short-term precipitation forecast based on the PERSIANN system and LSTM recurrent neural networks". In: *Journal of Geophysical Research: Atmospheres* 123.22 (2018), pp. 12–543.

[8] Md Saniul Alam and Aonghus McNabola. "Exploring the modeling of spatiotemporal variations in ambient air pollution within the land use regression framework: Estimation of PM10 concentrations on a daily basis". In: *Journal of the Air & Waste Management Association* 65.5 (2015), pp. 628–640.

[9] Shin Araki, Masayuki Shima, and Kouhei Yamamoto. "Spatiotemporal land use random forest model for estimating metropolitan NO2 exposure in Japan". In: *Science of The Total Environment* 634 (2018), pp. 1269–1277.

[10] V Athira, P Geetha, Rab Vinayakumar, and KP Soman. "Deepairnet: Applying recurrent networks for air quality prediction". In: *Procedia computer science* 132 (2018), pp. 1394–1403.

[11] Richard W Atkinson, H Ross Anderson, Jordi Sunyer, JON Ayres, Michela Baccini, Judith M Vonk, Azzedine Boumghar, Francesco Forastiere, Bertil Forsberg, Giota Touloumi, et al. "Acute effects of particulate air pollution on respiratory admissions: results from APHEA 2 project". In: *American journal of respiratory and critical care medicine* 164.10 (2001), pp. 1860–1866.

[12] Jorge Baño-Medina, Rodrigo Manzanas, and José Manuel Gutiérrez. "Configuration and intercomparison of deep learning neural models for statistical downscaling". In: *Geoscientific Model Development* 13.4 (2020), pp. 2109–2124.

[13] Bayerisches Landesamt für Digitalisierung, Breitband und Vermessung. *Digital Elevation Model*. https://geodatenonline.bayern.de/geodatenonline/seiten/dgm_info. 1996.

[14] Bayerisches Landesamt für Umwelt. *Soil Profile Database*. https://www.lfu.bayern.de/umweltdaten/index.htm. 2017.

[15] Rob Beelen, Gerard Hoek, Danielle Vienneau, Marloes Eeftens, Konstantina Dimakopoulou, Xanthi Pedeli, Ming-Yi Tsai, Nino Künzli, Tamara Schikowski, Alessandro Marcon, et al. "Development of NO2 and NOx land use regression models for estimating air pollution exposure in 36 study areas in Europe–the ESCAPE project". In: *Atmospheric Environment* 72 (2013), pp. 10–23.

[16] Sean D Beevers, Nutthida Kitwiroon, Martin L Williams, Frank J Kelly, H Ross Anderson, and David C Carslaw. "Air pollution dispersion models for human exposure predictions in London". In: *Journal of exposure science & environmental epidemiology* 23.6 (2013), pp. 647–653.

[17] Paula Branco, Luís Torgo, and Rita P. Ribeiro. "SMOGN: a Pre-processing Approach for Imbalanced Regression". In: *First International Workshop on Learning with Imbalanced Domains: Theory and Applications, LIDTA@PKDD/ECML 2017, 22 September 2017, Skopje, Macedonia*. Vol. 74. 2017, pp. 36–50.

[18] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[19] Cole Brokamp. "Land Use Random Forests for Estimation of Exposure to Elemental Components of Particulate Matter". PhD thesis. University of Cincinnati, 2016.

[20] Cole Brokamp, Roman Jandarov, MB Rao, Grace LeMasters, and Patrick Ryan. "Exposure assessment models for elemental components of particulate matter in an urban environment: A comparison of regression and random forest approaches". In: *Atmospheric Environment* 151 (2017), pp. 1–11.

[21] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[22] Bert Brunekreef and Stephen T Holgate. "Air pollution and health". In: *The lancet* 360.9341 (2002), pp. 1233–1242.

[23] Alexander G Buevich, Alexander N Medvedev, Alexander P Sergeev, Dmitry A Tarasov, Andrey V Shichkin, Marina V Sergeeva, and TB Atanasova. "Modeling of surface dust concentrations using neural networks and kriging". In: *AIP Conference Proceedings*. Vol. 1789. 1. 2016, p. 020004.

[24] Rich Caruana. "Multitask Learning". In: *Machine Learning* 28.1 (1997), pp. 41–75.

[25] Alexandre Champendal, Mikhail Kanevski, and Pierre-Emmanuel Huguenot. "Air pollution mapping using nonlinear land use regression models". In: *International Conference on Computational Science and Its Applications*. 2014, pp. 682–690.

[26] Ashesh Chattopadhyay, Ebrahim Nabizadeh, and Pedram Hassanzadeh. "Analog forecasting of extreme-causing weather patterns using deep learning". In: *Journal of Advances in Modeling Earth Systems* 12.2 (2020), e2019MS001958.

[27] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[28] Ling Chen, Yifang Ding, Dandan Lyu, Xiaoze Liu, and Hanyu Long. "Deep multi-task learning based urban air quality index modelling". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3.1 (2019), pp. 1–17.

[29] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, et al. "Xgboost: extreme gradient boosting". In: *R package version 0.4-2* 1.4 (2015), pp. 1–4.

[30] Weiyu Cheng, Yanyan Shen, Yanmin Zhu, and Linpeng Huang. "A neural attention model for urban air quality inference: Learning the weights of monitoring stations". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[31] François Chollet et al. *Keras*. https://keras.io. 2015.

[32] Ronan Collobert and Jason Weston. "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning". In: *Proceedings of the 25th International Conference on Machine Learning*. 2008, pp. 160–167.

[33] John Cook, Dana Nuccitelli, Sarah A Green, Mark Richardson, Bärbel Winkler, Rob Painting, Robert Way, Peter Jacobs, and Andrew Skuce. "Quantifying the consensus on anthropogenic global warming in the scientific literature". In: *Environmental research letters* 8.2 (2013), p. 024024.

[34] Thomas J Crowley. "Causes of climate change over the past 1000 years". In: *Science* 289.5477 (2000), pp. 270–277.

[35] Fuqiang Dai, Qigang Zhou, Zhiqiang Lv, Xuemei Wang, and Gangcai Liu. "Spatial prediction of soil organic matter content integrating artificial neural network and ordinary kriging in Tibetan Plateau". In: *Ecological Indicators* 45 (2014), pp. 184–194.

[36] Zihang Dai, Hanxiao Liu, Quoc Le, and Mingxing Tan. "Coatnet: Marrying convolution and attention for all data sizes". In: *Advances in Neural Information Processing Systems* 34 (2021).

[37] Saha Dauji and Ambavarafu Rafi. "Spatial interpolation of SPT with artificial neural network". In: *Engineering Journal* 25.2 (2021), pp. 109–120.

[38] Deutscher Wetterdienst. *Meteorological Data*. https://opendata.dwd.de/climate_environment/CDC/grids_germany/multi_annual/. 2020.

[39] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[40] Peter D Dueben and Peter Bauer. "Challenges and design choices for global weather and climate models based on machine learning". In: *Geoscientific Model Development* 11.10 (2018), pp. 3999–4009.

[41] Andrzej Dulny, Michael Steininger, Florian Lautenschlager, Anna Krause, and Andreas Hotho. "Evaluating the multi-task learning approach for land use regression modelling of air pollution". In: *Journal of Physics: Conference Series* 1834.1 (2021), p. 012004.

[42] David R Easterling, Gerald A Meehl, Camille Parmesan, Stanley A Changnon, Thomas R Karl, and Linda O Mearns. "Climate extremes: observations, modeling, and impacts". In: *science* 289.5487 (2000), pp. 2068–2074.

[43] Jonathan M Eden and Martin Widmann. "Downscaling of GCM-simulated precipitation using model output statistics". In: *Journal of Climate* 27.1 (2014), pp. 312–324.

[44] Marloes Eeftens, Rob Beelen, Kees de Hoogh, Tom Bellander, Giulia Cesaroni, Marta Cirach, Christophe Declercq, Audrius Dedele, Evi Dons, Audrey de Nazelle, et al. "Development of land use regression models for PM2. 5, PM2. 5 absorbance, PM10 and PMcoarse in 20 European study areas; results of the ESCAPE project". In: *Environmental science & technology* 46.20 (2012), pp. 11195–11205.

[45] Deng-Ping Fan, Ge-Peng Ji, Tao Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. "Pranet: Parallel reverse attention network for polyp segmentation". In: *International conference on medical image computing and computer-assisted intervention*. 2020, pp. 263–273.

[46] Junxiang Fan, Qi Li, Junxiong Hou, Xiao Feng, Hamed Karimian, and Shaofu Lin. "A spatiotemporal prediction framework for air pollution based on deep RNN". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4 (2017), p. 15.

[47] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. "Efficient and robust automated machine learning". In: *Advances in neural information processing systems* 28 (2015).

[48] Christopher B Field, Vicente Barros, Thomas F Stocker, and Qin Dahe. *Managing the risks of extreme events and disasters to advance climate change adaptation: special report of the intergovernmental panel on climate change*. 2012.

[49] Gregory Flato, Jochem Marotzke, Babatunde Abiodun, Pascale Braconnot, Sin Chan Chou, William Collins, Peter Cox, Fatima Driouech, Seita Emori, Veronika Eyring, et al. "Evaluation of climate models". In: *Climate change 2013: the physical science basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. 2014, pp. 741–866.

[50] Bastien François, Soulivanh Thao, and Mathieu Vrac. "Adjusting spatial dependence of climate model outputs with cycle-consistent adversarial networks". In: *Climate Dynamics* 57.11 (2021), pp. 3323–3353.

[51] Brian S Freeman, Graham Taylor, Bahram Gharabaghi, and Jesse Thé. "Forecasting air quality time series using deep learning". In: *Journal of the Air & Waste Management Association* 68.8 (2018), pp. 866–886.

[52] Masoud Ghahremanloo, Yannic Lops, Yunsoo Choi, and Bijan Yeganeh. "Deep Learning Estimation of Daily Ground-Level NO2 Concentrations From Remote Sensing Data". In: *Journal of Geophysical Research: Atmospheres* 126.21 (2021), e2021JD034925.

[53] X. Gibert, V. M. Patel, and R. Chellappa. "Deep Multitask Learning for Railway Track Inspection". In: *IEEE Transactions on Intelligent Transportation Systems* 18.1 (2017), pp. 153–164.

[54] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. 2016.

[55] Peter Grönquist, Chengyuan Yao, Tal Ben-Nun, Nikoli Dryden, Peter Dueben, Shigang Li, and Torsten Hoefler. "Deep learning for post-processing ensemble weather forecasts". In: *Philosophical Transactions of the Royal Society A* 379.2194 (2021), p. 20200092.

[56] Yoo-Geun Ham, Jeong-Hwan Kim, and Jing-Jia Luo. "Deep learning for multi-year ENSO forecasts". In: *Nature* 573.7775 (2019), pp. 568–572.

[57] David Hasenfratz, Olga Saukh, Christoph Walser, Christoph Hueglin, Martin Fierz, and Lothar Thiele. "Pushing the spatio-temporal resolution limit of urban air pollution maps". In: *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 2014, pp. 69–77.

[58] M R Haylock, N Hofstra, A M G Klein Tank, E J Klok, P D Jones, and M New. "A European daily high-resolution gridded data set of surface temperature and precipitation for 1950-2006". In: *Journal of Geophysical Research Atmospheres* 113.20 (2008), p. D20119.

[59] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning". In: *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. 2008, pp. 1322–1328.

[60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[61] Gerard Hoek, Rob Beelen, Kees de Hoogh, Danielle Vienneau, John Gulliver, Paul Fischer, and David Briggs. "A review of land-use regression models to assess spatial variation of outdoor air pollution". In: *Atmospheric environment* 42.33 (2008), pp. 7561–7578.

[62] Chin-Yu Hsu, Yu-Ting Zeng, Yu-Cheng Chen, Mu-Jean Chen, Shih-Chun Candice Lung, and Chih-Da Wu. "Kriging-based land-use regression models that use machine learning algorithms to estimate the monthly BTEX concentration". In: *International journal of environmental research and public health* 17.19 (2020), p. 6956.

[63] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. "Learning deep representation for imbalanced classification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 5375–5384.

[64] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. 2015, pp. 448–456.

[65] IPCC. *Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. 2014.

[66] D Jacob, BJJM Van den Hurk, Ulf Andrae, G Elgered, C Fortelius, LP Graham, SD Jackson, U Karstens, Chr Köpken, R Lindau, et al. "A comprehensive model inter-comparison study investigating the water budget during the BALTEX-PIDCAP period". In: *Meteorology and Atmospheric Physics* 77.1-4 (2001), pp. 19–43.

[67] Daniela Jacob. "A note to the simulation of the annual and inter-annual variability of the water budget over the Baltic Sea drainage basin". In: *Meteorology and Atmospheric Physics* 77.1-4 (2001), pp. 61–73.

[68] Ziyang Jiang, Tongshu Zheng, Mike Bergin, and David Carlson. "Improving spatial variation of ground-level PM2. 5 prediction with contrastive learning from satellite imagery". In: *Science of Remote Sensing* 5 (2022), p. 100052.

[69] Jovan Kalajdjieski, Eftim Zdravevski, Roberto Corizzo, Petre Lameski, Slobodan Kalajdziski, Ivan Miguel Pires, Nuno M Garcia, and Vladimir Trajkovik. "Air pollution prediction with multi-modal data and deep neural networks". In: *Remote Sensing* 12.24 (2020), p. 4142.

[70] Marilena Kampa and Elias Castanas. "Human health effects of air pollution". In: *Environmental pollution* 151.2 (2008), pp. 362–367.

[71] Klea Katsouyanni, Giota Touloumi, Evangelia Samoli, Alexandros Gryparis, Alain Le Tertre, Yannis Monopolis, Giuseppe Rossi, Denis Zmirou, Ferran Ballester, Azedine Boumghar, et al. "Confounding and effect modification in the short-term effects of ambient particles on total mortality: results from 29 European cities within the APHEA2 project". In: *Epidemiology* (2001), pp. 521–531.

[72] Alex Kendall, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7482–7491.

[73] Soo Kyung Kim, Sasha Ames, Jiwoo Lee, Chengzhu Zhang, Aaron C Wilson, and Dean Williams. "Massive scale deep learning for detecting extreme climate events". In: *Climate Informatics* (2017).

[74] Taehun Kim, Hyemin Lee, and Daijin Kim. "Uacanet: Uncertainty augmented context attention for polyp segmentation". In: *Proceedings of the 29th ACM International Conference on Multimedia*. 2021, pp. 2167–2175.

[75] Konstantin Kobs, Christian Schäfer, Michael Steininger, Anna Krause, Roland Baumhauer, Heiko Paeth, and Andreas Hotho. "Semi-Supervised Learning for Grain Size Distribution Interpolation". In: *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part VI*. 2021, pp. 34–44.

[76] Bartosz Krawczyk. "Learning from imbalanced data: open challenges and future directions". In: *Progress in Artificial Intelligence* 5.4 (2016), pp. 221–232.

[77] Hedi Katre Kriit, Johan Nilsson Sommar, Bertil Forsberg, Stefan Åström, Mikael Svensson, and Christer Johansson. "A health economic assessment of air pollution effects under climate neutral vehicle fleet scenarios in Stockholm, Sweden". In: *Journal of Transport & Health* 22 (2021), p. 101084.

[78] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[79] Florian Lautenschlager, Martin Becker, Konstantin Kobs, Michael Steininger, Padraig Davidson, Anna Krause, and Andreas Hotho. "OpenLUR: Off-the-shelf air pollution modeling with open features and machine learning". In: *Atmospheric Environment* 233 (2020), p. 117535.

[80] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), p. 436.

[81] Jason Jingshi Li, Boi Faltings, Olga Saukh, David Hasenfratz, and Jan Beutel. "Sensing the air we breathe—the OpenSense Zurich dataset". In: *Twenty-Sixth AAAI Conference on Artificial Intelligence*. 2012.

[82] Jiangeng Li, Xingyang Shao, and Rihui Sun. "A DBN-based deep neural network model with multitask learning for online air quality prediction". In: *Journal of Control Science and Engineering* 2019 (2019).

[83] Jin Li and Andrew D Heap. "Spatial interpolation methods applied in the environmental sciences: A review". In: *Environmental Modelling & Software* 53 (2014), pp. 173–189.

[84] Pengyong Li, Yuquan Li, Chang-Yu Hsieh, Shengyu Zhang, Xianggen Liu, Huanxiang Liu, Sen Song, and Xiaojun Yao. "TrimNet: learning molecular representation from triplet messages for biomedicine". In: *Briefings in Bioinformatics* 22.4 (2021), bbaa266.

[85] Yuncheng Li, Jifei Huang, and Jiebo Luo. "Using user generated online photos to estimate and monitor air pollution in major cities". In: *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*. 2015, p. 79.

[86] Zhanglin Li. "An enhanced dual IDW method for high-quality geospatial interpolation". In: *Scientific reports* 11.1 (2021), pp. 1–17.

[87] Zhiyuan Li, Xinning Tong, Jason Man Wai Ho, Timothy CY Kwok, Guanghui Dong, Kin-Fai Ho, and Steve Hung Lam Yim. "A practical framework for predicting residential indoor PM2. 5 concentration using land-use regression and machine learning methods". In: *Chemosphere* 265 (2021), p. 129140.

[88] Wu Liu, Xiaodong Li, Zuo Chen, Guangming Zeng, Tomás León, Jie Liang, Guohe Huang, Zhihua Gao, Sheng Jiao, Xiaoxiao He, et al. "Land use regression models coupled with meteorology to model spatial and temporal variability of NO2 and PM10 in Changsha, China". In: *Atmospheric Environment* 116 (2015), pp. 272–280.

[89] Weizhen Lu, Wenjian Wang, A.Y.T. Leung, Siu-Ming Lo, R.K.K. Yuen, Zongben Xu, and Huiyuan Fan. "Air pollutant parameter forecasting using support vector machines". In: *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*. Vol. 1. 2002, 630–635 vol.1.

[90] Xuying Ma, Ian Longley, Jennifer Salmond, and Jay Gao. "PyLUR: Efficient software for land use regression modeling the spatial distribution of air pollutants using GDAL/OGR library in Python". In: *Frontiers of Environmental Science & Engineering* 14.3 (2020), pp. 1–14.

[91] Detlev Majewski. "The Europa-Modell of the Deutscher Wetterdienst." In: *ECMWF Proc. "Numerical Methods in atmospheric models" Reading* 2 (1991), pp. 147–191.

[92] Douglas Maraun. "Bias correcting climate change simulations - a critical review". In: *Current Climate Change Reports* 2.4 (2016), pp. 211–220.

[93] Kane Maxwell, Mojtaba Rajabi, and Joan Esterle. "Spatial interpolation of coal properties using geographic quantile regression forest". In: *International Journal of Coal Geology* 248 (2021), p. 103869.

[94] Mercedes Medina-Ramon, Antonella Zanobetti, and Joel Schwartz. "The effect of ozone and PM10 on hospital admissions for pneumonia and chronic obstructive pulmonary disease: a national multicity study". In: *American journal of epidemiology* 163.6 (2006), pp. 579–588.

[95] Xia Meng, Li Chen, Jing Cai, Bin Zou, Chang-Fu Wu, Qingyan Fu, Yan Zhang, Yang Liu, and Haidong Kan. "A land use regression model for estimating the NO2 concentration in Shanghai, China". In: *Environmental research* 137 (2015), pp. 308–315.

[96] Sanaz Moghim and Rafael L Bras. "Bias correction of climate modeled temperature and precipitation using artificial neural networks". In: *Journal of Hydrometeorology* 18.7 (2017), pp. 1867–1884.

[97] Anna Molter. "XLUR: A land use regression wizard for ArcGIS Pro". In: *Journal of Open Source Software* 5.50 (2020), p. 2177.

[98] Denise R Montagne, Gerard Hoek, Jochem O Klompmaker, Meng Wang, Kees Meliefste, and Bert Brunekreef. "Land use regression models for ultrafine particles and black carbon based on short-term monitoring predict past spatial variation". In: *Environmental science & technology* 49.14 (2015), pp. 8712–8720.

[99] David W Morley and John Gulliver. "A land use regression variable generation, modelling and prediction tool for air pollution exposure assessment". In: *Environmental Modelling & Software* 105 (2018), pp. 17–23.

[100] Sheena Muttoo, Lisa Ramsay, Bert Brunekreef, Rob Beelen, Kees Meliefste, and Rajen N Naidoo. "Land use regression modelling estimating nitrogen oxides exposure in industrial south Durban, South Africa". In: *Science of the Total Environment* 610 (2018), pp. 1439–1447.

[101] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*. 2010, pp. 807–814.

[102] Muhammad Noor, Tarmizi bin Ismail, Shahid Ullah, Zafar Iqbal, Nadeem Nawaz, and Kamal Ahmed. "A non-local model output statistics approach for the downscaling of CMIP5 GCMs for the projection of rainfall in Peninsular Malaysia". In: *Journal of Water and Climate Change* (2019).

[103] Open Geospatial Consortium. "Definition of "geospatial"". In: *OGC Definitions Server*. 2018.

[104] OpenStreetMap contributors. *Planet dump retrieved from https://planet.osm.org*. https://www.openstreetmap.org. 2017.

[105] Ryota Otake, Jun Kurima, Hiroyuki Goto, and Sumio Sawada. "Deep learning model for spatial interpolation of real-time seismic intensity". In: *Seismological Society of America* 91.6 (2020), pp. 3433–3443.

[106] Heiko Paeth. "Postprocessing of simulated precipitation for impact research in West Africa. Part I: model output statistics for monthly data". In: *Climate Dynamics* 36.7-8 (2011), pp. 1321–1336.

[107] Baoxiang Pan, Kuolin Hsu, Amir AghaKouchak, and Soroosh Sorooshian. "Improving Precipitation Estimation Using Convolutional Neural Network". In: *Water Resources Research* 55.3 (2019), pp. 2301–2321.

[108] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. 2019, pp. 8024–8035.

[109] SE Perkins, AJ Pitman, NJ Holbrook, and J McAneney. "Evaluation of the AR4 climate models' simulated daily maximum temperature, minimum temperature, and precipitation over Australia using probability density functions". In: *Journal of climate* 20.17 (2007), pp. 4356–4376.

[110] Sahar Hadi Pour, Shamsuddin Shahid, Eun-Sung Chung, and Xiao-Jun Wang. "Model output statistics downscaling using support vector machine for the projection of spatial and temporal changes in rainfall of Bangladesh". In: *Atmospheric Research* 213 (2018), pp. 149–162.

[111] United Nations Environment Programme. *Making Peace with Nature*. 2021.

[112] R Core Team. *R: A Language and Environment for Statistical Computing*. 2022.

[113] Evan Racah, Christopher Beckham, Tegan Maharaj, Samira Ebrahimi Kahou, Mr Prabhat, and Chris Pal. "Extremeweather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events". In: *Advances in neural information processing systems* 30 (2017).

[114] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[115] Bharath Ramsundar, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding, and Vijay Pande. *Massively Multitask Networks for Drug Discovery*. 2015.

[116] Stephan Rasp, Michael S Pritchard, and Pierre Gentine. "Deep learning to represent subgrid processes in climate models". In: *Proceedings of the National Academy of Sciences* 115.39 (2018), pp. 9684–9689.

[117] Redlands, CA: Environmental Systems Research Institute. *ArcGIS Pro*. 2015.

[118] Jiawei Ren, Mingyuan Zhang, Cunjun Yu, and Ziwei Liu. "Balanced MSE for Imbalanced Visual Regression". In: *arXiv preprint arXiv:2203.16427* (2022).

[119] Khalil Rezaei, Bernard Guest, Anke Friedrich, Farajollah Fayazi, Muhammad Nakhaei, Ali Beitollahi, and Seyed Mahmoud Fatemi Aghda. "Feed forward neural network and interpolation function models to predict the soil and subsurface sediments distribution in Bam, Iran". In: *Acta Geophysica* 57.2 (2009), pp. 271–293.

[120] Rita P Ribeiro and Nuno Moniz. "Imbalanced regression and extreme value prediction". In: *Machine Learning* 109.9 (2020), pp. 1803–1835.

[121] Robert A Rohde and Richard A Muller. "Air pollution in China: mapping of concentrations and sources". In: *PloS one* 10.8 (2015), e0135749.

[122] David Rolnick, Priya L. Donti, Lynn H. Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, Alexandra Sasha Luccioni, Tegan Maharaj, Evan D. Sherwin, S. Karthik Mukkavilli, Konrad P. Kording, Carla P. Gomes, Andrew Y. Ng, Demis Hassabis, John C. Platt, Felix Creutzig, Jennifer Chayes, and Yoshua Bengio. "Tackling Climate Change with Machine Learning". In: *ACM Comput. Surv.* 55.2 (2022).

[123] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. 2015, pp. 234–241.

[124] Patrick H Ryan and Grace K LeMasters. "A review of land-use regression models for characterizing intraurban air pollution exposure". In: *Inhalation toxicology* 19.sup1 (2007), pp. 127–133.

[125] Zulfaqar Sa'adi, Shamsuddin Shahid, Eun-Sung Chung, and Tarmizi bin Ismail. "Projection of spatial and temporal changes of rainfall in Sarawak of Borneo Island using statistical downscaling of CMIP5 models". In: *Atmospheric research* 197 (2017), pp. 446–460.

[126] Apolline Saucy, Martin Röösli, Nino Künzli, Ming-Yi Tsai, Chloé Sieber, Toyib Olaniyan, Roslynn Baatjies, Mohamed Jeebhay, Mark Davey, Benjamin Flückiger, Rajen N Naidoo, Mohammed Aqiel Dalvie, Mahnaz Badpa, and Kees De Hoogh. "OP III – 5 Land use regression modelling of outdoor no2 and pm2.5 concentrations in three low-income areas of the urban western cape, south africa". In: *Exposure assessment*. 2018.

[127] Aleksandar Sekulić, Milan Kilibarda, Gerard Heuvelink, Mladen Nikolić, and Branislav Bajat. "Random forest spatial interpolation". In: *Remote Sensing* 12.10 (2020), p. 1687.

[128] Chao Shi and Yu Wang. "Non-parametric machine learning methods for interpolation of spatially varying non-stationary and non-Gaussian geotechnical properties". In: *Geoscience Frontiers* 12.1 (2021), pp. 339–350.

[129] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. "Deep learning for precipitation nowcasting: A benchmark and a new model". In: *Advances in neural information processing systems*. 2017, pp. 5617–5627.

[130] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. "Megatron-lm: Training multi-billion parameter language models using model parallelism". In: *arXiv preprint arXiv:1909.08053* (2019).

[131] N Clayton Silver and William P Dunlap. "Averaging correlation coefficients: Should Fisher's z transformation be used?" In: *Journal of Applied Psychology* 72.1 (1987), p. 146.

[132] Massimo Stafoggia, Tom Bellander, Simone Bucci, Marina Davoli, Kees de Hoogh, Francesca de'Donato, Claudio Gariazzo, Alexei Lyapustin, Paola Michelozzi, Matteo Renzi, Matteo Scortichini, Alexandra Shtein, Giovanni Viegi, Itai Kloog, and Joel Schwartz. "Estimation of daily PM10 and PM2.5 concentrations in Italy, 2013–2015, using a spatiotemporal land-use random-forest model". In: *Environment International* 124 (2019), pp. 170–179.

[133] Michael Steininger, Daniel Abel, Katrin Ziegler, Anna Krause, Heiko Paeth, and Andreas Hotho. "ConvMOS: Climate Model Output Statistics with Deep Learning". In: *Data Mining and Knowledge Discovery* (2022).

[134] Michael Steininger, Daniel Abel, Katrin Ziegler, Anna Krause, Heiko Paeth, and Andreas Hotho. "Deep Learning for Climate Model Output Statistics". In: *NeurIPS 2020 Workshop on Tackling Climate Change with Machine Learning* (2020).

[135] Michael Steininger, Konstantin Kobs, Padraig Davidson, Anna Krause, and Andreas Hotho. "Density-based weighting for imbalanced regression". In: *Machine Learning* 110.8 (2021), pp. 2187–2211.

[136] Michael Steininger, Konstantin Kobs, Albin Zehe, Florian Lautenschlager, Martin Becker, and Andreas Hotho. "MapLUR: Exploring a New Paradigm for Estimating Air Pollution Using Deep Learning on Map Images". In: *ACM Trans. Spatial Algorithms Syst.* 6.3 (2020).

[137] Lucas Stephens et al. "Archaeological assessment reveals Earth's early transformation through land use". In: *Science* 365.6456 (2019), pp. 897–902.

[138] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[139] DA Tarasov, AG Buevich, AP Sergeev, and AV Shichkin. "High variation topsoil pollution forecasting in the Russian Subarctic: Using artificial neural networks combined with residual kriging". In: *Applied Geochemistry* 88 (2018), pp. 188–197.

[140] Luís Torgo, Rita P Ribeiro, Bernhard Pfahringer, and Paula Branco. "Smote for regression". In: *Portuguese conference on artificial intelligence*. 2013, pp. 378–389.

[141] G Touloumi, K Katsouyanni, D Zmirou, J Schwartz, C Spix, A Ponce de Leon, A Tobias, P Quennel, D Rabczenko, L Bacharova, et al. "Short-term effects of ambient oxidant exposure on mortality: a combined analysis within the APHEA project". In: *American journal of epidemiology* 146.2 (1997), pp. 177–185.

[142] Hasheel Tularam, Lisa F Ramsay, Sheena Muttoo, Bert Brunekreef, Kees Meliefste, Kees de Hoogh, and Rajen N Naidoo. "A hybrid air pollution/land use regression model for predicting air pollution concentrations in Durban, South Africa". In: *Environmental Pollution* 274 (2021), p. 116513.

[143] Jesper E Van Engelen and Holger H Hoos. "A survey on semi-supervised learning". In: *Machine Learning* 109.2 (2020), pp. 373–440.

[144] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. 2009.

[145] Thomas Vandal, Evan Kodra, Sangram Ganguly, Andrew Michaelis, Ramakrishna Nemani, and Auroop R Ganguly. "Deepsd: Generating high resolution climate change projections through single image super-resolution". In: *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*. 2017, pp. 1663–1672.

[146] Pilar Vizcaino and Carlo Lavalle. "Development of European NO 2 Land Use Regression Model for present and future exposure assessment: Implications for policy analysis". In: *Environmental Pollution* 240 (2018), pp. 140–154.

[147] Meng Wang, Rob Beelen, Tom Bellander, Matthias Birk, Giulia Cesaroni, Marta Cirach, Josef Cyrys, Kees de Hoogh, Christophe Declercq, Konstantina Dimakopoulou, et al. "Performance of multi-city land use regression models for nitrogen dioxide and fine particles". In: *Environmental health perspectives* 122.8 (2014), p. 843.

[148] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. "Learning to model the tail". In: *Advances in Neural Information Processing Systems*. 2017, pp. 7029–7039.

[149] Oliver Watt-Meyer, Noah D Brenowitz, Spencer K Clark, Brian Henn, Anna Kwa, Jeremy McGibbon, W Andre Perkins, and Christopher S Bretherton. "Correcting weather and climate models by machine learning nudged historical simulations". In: *Geophysical Research Letters* 48.15 (2021), e2021GL092555.

[150] Frank Wilcoxon. "Individual Comparisons by Ranking Methods". In: *Biometrics Bulletin* 1.6 (1945), pp. 80–83.

[151] Kathrin Wolf, Josef Cyrys, Tatiana Harciníková, Jianwei Gu, Thomas Kusch, Regina Hampel, Alexandra Schneider, and Annette Peters. "Land use regression modeling of ultrafine particles, ozone, nitrogen oxides and markers of particulate matter pollution in Augsburg, Germany". In: *Science of the Total Environment* 579 (2017), pp. 1531–1540.

[152] Pei-Yi Wong, Chin-Yu Hsu, Jhao-Yi Wu, Tee-Ann Teo, Jen-Wei Huang, How-Ran Guo, Huey-Jen Su, Chih-Da Wu, and John D Spengler. "Incorporating land-use regression into machine learning algorithms in estimating the spatial-temporal variation of carbon monoxide in Taiwan". In: *Environmental Modelling & Software* 139 (2021), p. 104996.

[153] Pei-Yi Wong, Hsiao-Yun Lee, Yu-Cheng Chen, Yu-Ting Zeng, Yinq-Rong Chern, Nai-Tzu Chen, Shih-Chun Candice Lung, Huey-Jen Su, and Chih-Da Wu. "Using a land use regression model with machine learning to estimate ground level PM2. 5". In: *Environmental Pollution* 277 (2021), p. 116846.

[154] Jiansheng Wu, Jiacheng Li, Jian Peng, Weifeng Li, Guang Xu, and Chengcheng Dong. "Applying land use regression model to estimate spatial variation of PM2. 5 in Beijing, China". In: *Environmental Science and Pollution Research* 22.9 (2015), pp. 7045–7061.

[155] Xinghan Xu and Minoru Yoneda. "Multitask air-quality prediction based on LSTM-autoencoder model". In: *IEEE transactions on cybernetics* 51.5 (2019), pp. 2577–2586.

[156] Rui Yan, Jiaqiang Liao, Jie Yang, Wei Sun, Mingyue Nong, and Feipeng Li. "Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering". In: *Expert Systems with Applications* 169 (2021), p. 114513.

[157] Yuzhe Yang, Kaiwen Zha, Yingcong Chen, Hao Wang, and Dina Katabi. "Delving into Deep Imbalanced Regression". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. 2021, pp. 11842–11851.

[158] Janni Yuval, Paul A O'Gorman, and Chris N Hill. "Use of neural networks for stable, accurate and physically consistent parameterization of subgrid atmospheric processes with good performance at reduced precision". In: *Geophysical Research Letters* 48.6 (2021), e2020GL091363.

[159] Chao Zhang, Junchi Yan, Changsheng Li, Hao Wu, and Rongfang Bie. "End-to-end learning for image-based air quality level estimation". In: *Machine Vision and Applications* 29.4 (2018), pp. 601–615.

[160] Yu Zhang and Qiang Yang. "An overview of multi-task learning". In: *National Science Review* 5.1 (2018), pp. 30–43.

[161] Tongshu Zheng, Michael H Bergin, Shijia Hu, Joshua Miller, and David E Carlson. "Estimating ground-level PM2. 5 using micro-satellite images by a convolutional neural network and random forest approach". In: *Atmospheric Environment* 230 (2020), p. 117451.

[162] Di Zhu, Ximeng Cheng, Fan Zhang, Xin Yao, Yong Gao, and Yu Liu. "Spatial interpolation using conditional generative adversarial neural networks". In: *International Journal of Geographical Information Science* 34.4 (2020), pp. 735–758.