Monitoring and Assessment of Air Quality with Semiconducting Metal Oxides

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Arne Kobald

aus Berlin

Tübingen

2025

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	17. Juni 2025
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter	Prof. Dr. Udo Weimar
2. Berichterstatter	Prof. Dr. Marcus Scheele

Declaration of Authorship

I hereby declare that I am the sole author of this Doctoral Thesis and that I have not used any sources other than those listed in the bibliography and identified as references. I further declare that I have not submitted this thesis at any other institution in order to obtain a degree.

This thesis was worked out and written at the Institute for Physical and Theoretical Chemistry at the Eberhard Karls University Tübingen under supervision of Prof. Dr. Udo Weimar and Dr. Nicolae Bârsan between May, 2019 and February, 2025.

Date

Arne Kobald

„All models are wrong,
but some are useful.“

— George E. P. Box

Contents

Abstract

Air quality monitoring is crucial for understanding pollutant dynamics and their impacts on health and the environment. The number of people living in urban areas has surpassed the number in rural areas and anthropogenic pollution has become more significant than natural sources. As a consequence, many guidelines for air quality are brought about and call for strict control. To this day, reference instruments are large and cost intensive, limiting the available coverage with public measurement stations. While some pollutants like ozone (O_3) are largely influenced by seasonal patterns, others are locally more confined to their sources like nitrous oxides (NOx) at busy traffic sections, requiring a larger spacial measurement resolution.

This research has been focused on leveraging gas sensors based on temperature modulated Semiconducting Metal Oxides (SMOX) in combination with advanced chemometric algorithms to provide small, cheap, and simple to use devices. To this end, a miniaturized commercial SMOX gas sensor array (SGP30, Sensirion AG, Switzerland) has been operated under controlled laboratory conditions in the first phase of this thesis. Multiple regressors have been optimized and trained to predict the concentration of the pollutant gases carbon monoxide (CO), nitrogen dioxide (NO_2), ozone (O_3), and sulfur dioxide (SO_2) in random gaseous mixtures in the range of guidelines published by the World Health Organization (WHO). Simultaneously, the concentration of other relevant gaseous species has been varied randomly, including relative humidity (RH) at 25 °C – an interfering gas known to influence the sensitivity of SMOX gas sensors, and a background of hydrogen (H_2) and methane (CH_4).

The group of utilized algorithms for quantification in this work included in order of increasing complexity and number of trainable parameters: ridge regression – a type of well established regularized multilinear regression, XGBoost (EXtreme Gradient Boost) – an ensemble of classification and regression trees winning multiple machine learning challenges, and Convolutional Neural Networks (CNNs) – an architecture of artificial neural networks often employed in computer vision. After careful data curation and handling, the Mean Relative Error (MRE) in a test subset – unrelated to that of the fitting process – was below 10 % on average for the best CNN.

Due to this successfully demonstrated feasibility of the sensing approach in the initial work, the same concept has been applied to real outdoor air with the goal of detecting atmospheric ozone on the campus Morgenstelle in Tübingen. The readings of a nearby public environmental measurement station by the UBA („Umweltbundesamt“, A German environment protection agency) have been used for training, validating, and testing the group of chemometric models. The higher MRE above 20 % compared to the laboratory experiment could be caused by a number of factors including the uncontrollable environment with harsh changes of ambient temperature and relative humidity. The weighted coefficients of the optimized ridge regression have been interpreted in terms of the importance of different temperature plateaus for the ozone prediction. Ultimately, a temperature cycle with a higher start and end temperature has been defined, expected to yield robust and information-rich sensor readings in a long term experiment for more than one year during this study.

The performance of the models resulting from the long term experiment turned out underwhelming. The new temperature cycle was found unsuitable for the quantification of ozone even for a shorter duration, limiting the assessment of long term stability or evaluation of sensor aging.
The introduction of the SGP30's successor SGP40 and the announced end-of-life status of the former, motivated the use and comparison of both sensor iterations. A versatile sensing platform including hard- and software around the ESP32 (Espressif Systems, China) microcontroller has been developed as part of this thesis to ease the use and investigation of commercial digital gas sensors not limited to those of Sensirions or this work. Subsequently, the SGP40 has been found to be a suitable drop-in replacement and has further been implemented in the work group's reference sensor unit as well as a novel modular and self contained sensing platform in collaboration with *AO Action*, the „eNose platform". The latter proved a necessary development, allowing the strategic placement of sensor nodes close to reference stations.
The current study carried out a SHapley Additive Explanations (SHAP) analysis to truly inspect the importance of different sensing temperatures especially across the different complex models that are often treated as black boxes. This methodology explains the importance of different features and attributes numeric contributions to the prediction of analyte gases to distinct operation temperatures or sensing materials. The best CNN was found to extract a large number of meaningful features and generalize well to unseen data in both laboratory and real ambient conditions. The ridge regression yielded more noisy information, manifested in higher errors. The XGBoost regressor has based the quantification on very few features, not able to support the desired accuracy. Based on this analysis, relevant features of the temperature cycle have been selected to train a slimmer CNN model – with half the trainable parameters and simpler to optimize – providing comparable accuracy for the prediction of ozone.
Further, this work has brought up a Raspberry Pi module to operate in-house fabricated or commercial analog sensors in a similar fashion with a modulation of the heater temperature to expand the pool of available sensing materials to be screened for applications. The so called „SenPi" unit has been put to test in the prediction of ozone in winter 2024. The single sensors could not been trained to predict O_3 with the desired accuracy in this short duration of rather small ambient concentrations. Nonetheless, the sensing shield has been proven to work well in modulating the operation temperature of (miniaturized) analog sensors and measure their resulting resistance over several orders of magnitude.
The final demonstrator and prototype in the work at hand, the „eNose platform", that is not limited to air quality measurements, but enables applications like breath analysis or quality control, has successfully been tested in Tübingen for the prediction of O_3. Another unit has been deployed in the city center of Trento, Italy, next to a public reference module for the last three months of 2024. Here, the direct proximity allowed building a predictive model for the concentrations of not only O_3, but also NO_2 and SO_2. The predictive performance with an average MRE around 20 % validated the overall measurement scheme developed here. The approach is applicable in more than one place, during multiple seasons and for different pollutant gases.

In summary, the applicability of temperature modulated SMOX gas sensors has been demonstrated for the quantification of pollutant gases when coupled with capable chemometric algorithms. A SHAP analysis has been applied to explain how CNN models outperformed other established chemometric algorithms. The importance of low temperature measurements has been highlighted, especially for the prediction of tropospheric ozone in multiple locations. Soft- and hardware frameworks for the operation of digital and analog gas sensors have been developed in the frame of this thesis, that will continue to aid gas sensor research. The standalone measurement instrument „eNose Platform “is in use for tasks like breath analysis, food and process control and in various air quality measurement setups around the globe.

Zusammenfassung

Die Überwachung der Luftqualität ist entscheidend für das Verständnis der Schadstoffdynamik und ihrer Auswirkungen auf Gesundheit und Umwelt. Heute ist die Zahl der Menschen, die in städtischen Gebieten leben größer als die derer in ländlichen Gebieten und die anthropogene Verschmutzung ist bedeutender geworden als natürliche Quellen. Dies hat zur Folge, dass zahlreiche Richtlinien für die Luftqualität erlassen wurden, die eine strenge Kontrolle erfordern. Bis heute sind die Referenzmessgeräte groß und kostenintensiv, was die Abdeckung mit öffentlichen Messstationen einschränkt. Während einige Schadstoffe wie Ozon (O_3) weitgehend durch saisonale und geographische Faktoren beeinflusst werden, sind andere stark lokalisiert wie zum Beispiel Stickoxide (NOx) an stark befahrenen Straßenabschnitten, was eine größere räumliche Messauflösung erfordert.

Diese Ausarbeitung konzentrierte sich auf die Nutzung von Gassensoren auf der Grundlage von temperaturmodulierten halbleitenden Metalloxiden (SMOX) in Kombination mit fortschrittlichen chemometrischen Algorithmen, um kleine, kostengünstige und und einfach zu handhabende Geräte bereitzustellen. Zu diesem Zweck wurde ein miniaturisiertes kommerzielles SMOX-Gassensormodul (SGP30, Sensirion AG, Schweiz) in der ersten Phase der Arbeit unter kontrollierten Laborbedingungen betrieben. Mehrere Regressoren wurden optimiert und trainiert, um die Konzentration der Schadstoffgase Kohlenmonoxid (CO), Stickstoffdioxid (NO_2), Ozon (O_3) und Schwefeldioxid (SO_2) in Gasgemischen im Bereich der von der Weltgesundheitsorganisation (WHO) veröffentlichten Richtlinien zu quantifizieren. Zeitgleich wurde die Konzentration anderer relevanter Gase randomisiert verändert, einschließlich der relativen Luftfeuchtigkeit (RH) bei 25 °C – ein Störgas, das bekanntermaßen die Empfindlichkeit von SMOX-Gassensoren beeinflussen kann, sowie eines Hintergrunds aus Wasserstoff (H_2) und Methan (CH_4).

Die Gruppe der für die Quantifizierung verwendeten Algorithmen umfasste nach zunehmender Komplexität und Anzahl der zu optimierenden Parameter: Ridge Regression – eine etablierte Variante regularisierter multilinearer Regression, XGBoost (EXtreme Gradient Boost) – ein Ensemble von Klassifikations- und Regressionsbäumen, die sich in mehreren Wettbewerben des maschinellen Lernens durchgesetzt hat und Convolutional Neural Networks (CNNs) – faltende neuronale Netze, die häufig in Computer Vision eingesetzt wird. Nach sorgfältiger Datenkuration und -bearbeitung lag der mittlere relative Fehler (MRE) im Testdatensatz, unabhänig von jenem des Optimierens, im Durchschnitt unter 10 % für das beste CNN.

Aufgrund der erfolgreich nachgewiesenen Machbarkeit dieser Analysemethode im ersten Abschnitt dieser Studie wurde das gleiche Konzept auf die auf reale Außenluft angewandt, mit dem Ziel, atmosphärisches Ozon auf dem Campus Morgenstelle in Tübingen zu bestimmen. Die Messwerte einer nahe gelegenen öffentlichen Messstation des UBA („Umweltbundesamt“) wurden für das Training, die Validierung und den Test der chemometrischen Modelle verwendet.

Der höhere MRE-Wert von über 20 % im Vergleich zum Laborversuch könnte auf eine Reihe von Faktoren zurückzuführen sein, darunter die unkontrollierbare Umgebung mit starken Schwankungen der Lufttemperatur und relativen Luftfeuchtigkeit. Die gewichteten Koeffizienten der optimierten Ridge-Regression wurden im Hinblick auf die Bedeutung der verschiedenen Temperaturplateaus für die Ozonvorhersage interpretiert. Daraus wurde ein Temperaturzyklus mit einer höheren Start- und Endtemperatur abgeleitet, der in einem Langzeitexperiment über mehr als ein Jahr während dieser Arbeit robuste und informationsreiche Sensormesswerte liefern sollte. Die Aussagekraft der Modelle, die sich aus dem Langzeitexperiment ergaben, erwies sich als nicht überzeugend und der neue Temperaturmodus zeigte sich unangebracht für die Vorhersage von Ozon. Dies bestätigte sich auch für eine kürzerze Experimentdauert, was eine Aussage über die Langzeitstabilität und Sensoralterung erschwert.
Die Einführung des SGP30-Nachfolgers SGP40 und die angekündigte Auslaufphase ersterens begründet die Verwendung und den Vergleich beider Sensor-Iterationen. Dazu wurde eine flexible Sensorplattform mit Hard- und Software rund um den ESP32 (Espressif Systems, China) im Rahmen dieser Arbeit entwickelt, um die Nutzung und Untersuchung kommerzieller digitaler Gassensoren zu erleichtern, die nicht auf jene von Sensirion oder die vorliegende Studie beschränkt sind. Der SGP40 hat sich als geeigneter Ersatz erwiesen und wurde unter anderem in die Referenzsensoreinheit der Arbeitsgruppe des Authors sowie in eine neuartige, modulare und eigenständige Sensorplattform in Zusammenarbeit mit *AO Action*, die „eNose Platform“, implementiert. Letztere erwies sich als eine notwendige Entwicklung für die vorliegende Arbeit, die die strategische Platzierung von Sensoreinheiten in der Nähe von Referenzstationen ermöglicht.
Diese Studie umfasste eine SHapley Additive exPlanation (SHAP) Analyse, um die Bedeutung der unterschiedlichen Sensortemperaturen auch bei komplexen Modellen, die oft als „Black Box“ behandelt werden, zu untersuchen. Diese Methodik erklärt die Bedeutung der verschiedenen Merkmale und ordnet die numerischen Beiträge zur Vorhersage der analysierten Gase den verschiedenen Betriebstemperaturen oder Sensormaterialien zu. Es wurde gezeigt, dass das beste neuronale Netz eine große Anzahl von aussagekräftigen Merkmalen extrahiert und sich sowohl im Labor als auch unter realen Umgebungsbedingungen gut auf neue ungesehene Daten verallgemeinern lässt. Die Ridge Regression lieferte mehr Rauschen, was sich in höheren Fehlern wiederspiegelte. Die XGBoost-Regression hat die Quantifizierung auf sehr wenige Merkmale gestützt, die nicht in der Lage waren, die gewünschte Genauigkeit zu erreichen.
Auf der Grundlage dieser Analyse wurden relevante Merkmale des Temperaturzyklus' ausgewählt, um ein schlankeres CNN-Modell – mit der Hälfte der trainierbaren Parameter und damit einfacher zu optimieren – zu trainieren, das eine vergleichbare Genauigkeit für die Vorhersage von Ozon bietet.
Schließlich entstand aus dieser Arbeit ein Modul für einen Raspberry-Pi, mit dem selbst hergestellte oder kommerzielle analoge Sensoren in ähnlicher Weise mit einer Modulation der Heiztemperatur betrieben werden können, um den Pool an verfügbaren Sensormaterialien zu erweitern, die für verschiedenste Anwendungen geprüft werden können. Das so genannte „SenPi“-Modul wurde für die Vorhersage von Ozon im Winter 2024 getestet. Die beiden Sensoren konnten nicht darauf trainiert werden, O_3 mit der gewünschten Genauigkeit in dieser kurzen Zeitspanne mit eher geringen Umgebungskonzentrationen vorherzusagen.

Nichtsdestotrotz konnte gezeigt werden, dass das Sensormodul gut funktioniert, um die Betriebstemperatur von (miniaturisierten) analogen Sensoren zu modulieren und ihren resultierenden Widerstand über mehrere Größenordnungen zu messen. Der endgültige Demonstrator und Prototyp in dieser Untersuchung – die „eNose-Plattform“, die nicht auf die Messung der Luftqualität beschränkt ist, sondern auch Anwendungen in der Atemagasnalyse oder Qualitätskontrolle finden kann, wurde in Tübingen erfolgreich für die Vorhersage von O_3 getestet. Ein weiteres Gerät wurde in den letzten drei Monaten des Jahres 2024 im Stadtzentrum von Trient, Italien, neben einem öffentlichen Referenzmodul aufgestellt. Hier ermöglichte die unmittelbare Nähe die Erstellung eines Vorhersagemodells für die Konzentrationen von nicht nur O_3, sondern auch von NO_2 und SO_2. Die Vorhersageleistung mit einem durchschnittlichen MRE von etwa 20 % validierte den gesamten hier erarbeiteten Analyseprozess, der an mehr als einem Ort, zu mehreren Jahreszeiten und für mehrere Schadgase anwendbar ist. Zusammenfassend lässt sich sagen, dass die Anwendbarkeit von temperaturmodulierten SMOX-Gassensoren für die Quantifizierung von Schadgasen in Verbindung mit leistungsfähigen chemometrischen Algorithmen anhand mehrerer Beispiele demonstriert wurde. Eine SHAP Analyse wurde angewandt um zu erklären warum CNN Modelle performanter als andere etablierte Regressoren sind. Die Notwendigkeit von Niedertemperaturmessungen wurde vor allem für die Vorhersage von atmosphärischem Ozon an verschiedenen Orten aufgezeigt. Für diese Arbeit wurde eine Reihe von Soft- und Hardware-Frameworks für den Betrieb von analogen und digitalen Gassensoren entwickelt, die auch in der Zukunft Anwendung in der Forschung an Gassensoren finden wird.

1 Introduction and Motivation

The real-time monitoring of air quality and pollution is of increasing interest world wide due to its importance on health and the environment. The World Health Organization (WHO) recognizes air pollution as „the single biggest environmental threat to human health“ [1]. As a consequence they publish guidelines for air quality – both indoors and outdoors – with recommend short- and long-term levels. The classical air pollutants are CO, NO_2, O_3, SO_2, and Particulate Matter (PM) grouped into aerodynamic diameters smaller then 2.5 and 10 µm – $PM_{2.5}$ and PM_{10} respectively. While sources of air pollution are both natural and man-made, the latter is becoming dominant with the globally ever progressing industrialization [2]. The majority of anthropogenic particulate matter originates from combustion and production processes with elemental carbon being the main constituent. The other main pollutants are well defined gaseous species: CO is a colorless and odorless gas that is highly toxic as it binds strongly to hemoglobin and inhibits oxygen transport. Its main source is incomplete combustion for instance in oxygen deprived atmospheres. NO_2 and SO_2 predominantly stem from combustion of fossil fuels and are main contributors to acid rain and causing respiratory and cardiovascular diseases. Nitrous oxide (NO_2) may also be measured jointly with nitric oxide (NO) expressed as NO_x. Tropospheric (ground-level) O_3 is a secondary pollutant being the product of photochemical reactions of oxygen and the other pollutants which act as precursors. In contrast to stratospheric – which blocks harmful ultraviolet radiation in the ozone-oxygen cycle – it causes reduced lung functionality and even interferes with the photosynthesis of plants [3]. Ground-level O_3 concentrations vary less dramatically on a regional scale and are mainly influenced by seasonal changes [1]. Tropospheric ozone levels have been increasing steadily since the 20th century and yearly peak concentrations have shifted from spring or winter to summer, highlighting the complex interplay of combustion emission, irradiation and heat [4].

A number of websites and resources exist with maps of reported pollutant concentrations outdoors and associated recommendations regarding behavior like outdoor activities. For Europe, such a service is the „European Air Quality Index“ [5]. Contributions towards air quality data are made by governments, organizations, and individuals alike (e. g. in citizen science projects). However, the density and coverage of measurement stations could be greatly improved and extended to indoor air quality if smaller, less expensive solutions existed. The ISO standard for the quantification of ambient O_3 for instance is the ultraviolet photometric method relying on large analytical instruments [6]. Gas sensors based on Semiconducting Metal Oxide (SMOX) on the other hand, combine excellent sensitivity and broad application range [7]. The first report of a gas sensitive metal oxide was published by Heiland, describing the conductivity change of ZnO depending on the adsorbed oxygen concentration [8]. The first commercially SMOX gas sensors were developed by Taguchi and produced by Figaro [9].

The main benefit of SMOX gas sensors compared to other analytical instrumentation is the ability to miniaturize them for small size and low cost [10]. One of the biggest drawbacks of this type of gas sensor is the limited selectivity and cross-sensitivity to humidity [11]. While much work is invested in the design of selective materials – e. g. by doping or loading [12] – this property makes them suitable for a wide range of analytes. In order to derive qualitative or quantitative information about individual gas species in a mixture of analytes or in a complex matrix, an array of sufficiently different gas sensors may be combined into an array. Such a unit, merging multiple desirable features, is also called an electronic Nose – or eNose for short – when combined with signal processing [13]. Applications are manifold and include but are not limited to fire safety [14], breath analysis [15], and quality control [16]. The complexity of a sensor array can be increased by the use of different sensing principles [17]. Another approach to generate a higher information content without added size or cost is the modulation of the operation temperature of SMOX sensors [18]. The transient response of the resistance due to altered rates of reaction was first used for qualitative analysis to classify different gas molecules. Heilig et al. were able to use a single SnO_2 sensors to quantify CO and NO_2 in air [19]. A similar scheme is often employed in commercial gas sensors, e. g. in the TGS3870 [20] targeting CO and CH_4 prediction during the cold and hot phase, respectively. The multidimensional reading of a single modulated sensor is sometimes called a virtual sensor array as the output pattern contains multiple partially independent elements [13]. The field of chemometrics describes the use of mathematical models or algorithms to extract chemical information from multivariate data of experiments or instruments such as a sensor array, well aware of statistics and noise or uncertainties [21]. Among the plethora of models are Artificial Neural Networks (ANNs) in varying flavors. Convolutional Neural Networks (CNNs) are especially well adopted due to their wide use in Computer Vision (CV) and one of the first applications to predict gases was „GasNet“ reported by Peng et al. for the classification of gaseous species based on a gas sensor array of eight SMOX gas sensors [22]. Besides others, this concept has been further adapted by Robin et al. for the accurate detection of gas concentrations relevant in Indoor Air Quality (IAQ) with inputs from temperature-cycled SMOX gas sensors [23]. Most related work focuses on classification problems like the identification of gases[24–26], detection of the presence above a certain threshold, regression with discrete concentrations [27, 28] or recognizing certain (fault) states [29]. However, other studies report promising results about continuous concentrations in complex mixtures [30].

Though most quantitative models reveal some kind of regression parameters, it is difficult to compare them between different models or even architectures. The importance assigned to different input features is not directly accessible either, as the original inputs are inherently extracted or prioritized differently and especially large and complex models are seen as a black box. Lloyd Shapley introduced a measure to quantify the contribution or payout of individual players in cooperative game theory [31] and the Shapley value can be used to explain how the output (≡ payout) of a predictive model (≡ game) is controlled by the input features (≡ players). Lundberg et al. used and extended this formalism to SHapley Additive exPlanations (SHAP) values making model specific approximations to assure computational feasibility [32]. Santos et al. applied their framework to ANNs forecasting ozone concentrations and revealed that solar irradiance becomes especially important when increasing the forecast horizon [33]. Betancourt et al. validate their high-resolution mapping of tropospheric ozone

on a global level with SHAP values matching commonly accepted knowledge [34]. The most important factors governing the O_3 concentration are shown to be relative altitude and latitude.
This thesis is devoted to the quantization of pollutant gases for outdoor air quality by means of temperature modulated SMOX gas sensors and chemometric methods to provide cost and size efficient tools to make monitoring more accessible. A number of hard- and software frameworks were developed to ease the screening, operation, and characterization of analog and digital SMOX gas sensors and will be introduced in chapter 3,*Material and Methods*. Different chemometric algorithms including ridge regression, EXtreme Gradient Boosting (XGBoost) regression, and CNNs will be introduced in section 3.3, *Data Processing*. Starting from a laboratory pilot study with fully random and continuous mixtures of pollutant gases, these methods have been applied to various data sets acquired in real outdoor air with the majority revolving around the prediction of tropospheric ozone in Tübingen with commercial, digital multipixel Micro-Electro-Mechanical Systems (MEMS) gas sensors. SHAP analyses were carried out to understand and compare the feature selection and their importance for each applied model. The findings could be used to downsample the data and train or obtain slimmer CNN models. Lastly, the study has been widened to not only validate the newly developed standalone devices, but also test them in a different city – namely in Trento, Italy – for the detection of multiple gaseous species.

2 Fundamentals and Literature

2.1 SMOX Gas Sensors

Gas sensors or detectors are devices that detect or quantify the presence of certain analyte gases. They come in a large number or types or sizes; depending on the application scenario with different benefits and shortcomings. In general, gas sensors can be classified according to their receptor and transducer principle. The chemical information is transformed by the receptor into an energy (difference) measurable by the transducer [35]. Often discussed properties are selectivity, the ability to produce a signal independent of interferants, i. e. the presence of other gases [36], and sensitivity, the magnitude of a sensor signal change for a given difference of the stimulus – that is the slope of the calibration curve at a given concentration. The SMOX gas sensors encountered in this work are chemoresistive (sometimes also called chemiresistive). The readout of the sensor is electrical resistance of a semiconducting metal oxide measured between electrodes, which changes due to chemical surface reactions. The properties of a gas sensor always have to be seen and discussed as a whole, including besides the sensing material also substrate, electrodes, readout electronics, and packaging. Dependent on the synthesis route or distinct treatment of the powders different properties like the defect concentration [37] or presence of certain surface groups (e. g. hydroxyl group coverage [38]) may be encountered and tuned. Doping or loading with noble metals is often used to enhance stability, selectivity or sensitivity [12].
Sensors may be constructed from single crystals, thin or thick films. Usually the measured property that changes upon gas exposure is resistance. While single crystals have a very defined surface, thick film gas sensors show the largest sensor signals due to multiple inter-granular Schottky barriers [39]. A great body of literature exists, describing and modeling the reception and transduction processes of gas sensors based on SMOX materials: [7, 40–44]. The generally accepted model describes a porous layer of SMOX grains and considers the creation or cancellation of electrically active surface species. Figure 2.1 shows the model for an n-type SMOX in an atmosphere that contains O_2 as is the case in most applications. Depending on temperature, material and surface, different concentrations of electronegative oxygen species ${O_\beta}^{\alpha-}$ adsorb, localizing and trapping electrons at the surface. The resulting depletion layer with a decreased concentration of majority charge carriers – electrons – gives rise to an upward band bending depicted in the energy band diagram on the bottom. For sufficiently large grains – relative to the Debye length – the charge carrier concentration in the bulk and the position of the Fermi level stay constant. Consequently, the electrical resistance is governed by the energetic height of the back to back Schottky barriers. Out of a large variety of target gases, reducing CO presents an important model gas, that is catalytically oxidized to form thermodynamically stable carbon dioxide CO_2 readily, as depicted on the right side. The decrease of negatively charged ionosorbed oxygen surface species or the creation of oxygen vacancies causes the electrical signal, i. e. reduced resistance for n-type semiconductors due to a lowered energy barrier.

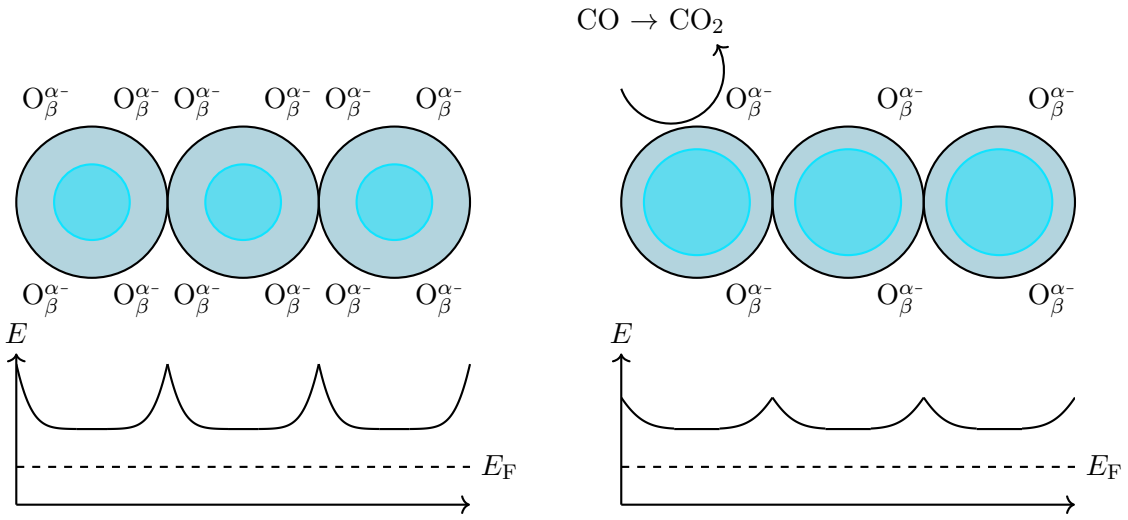


Figure 2.1: Schematic reception/sensing of a porous SMOX gas sensor in a depletion layer controlled conduction mechanism. CO is catalytically oxidized to CO_2

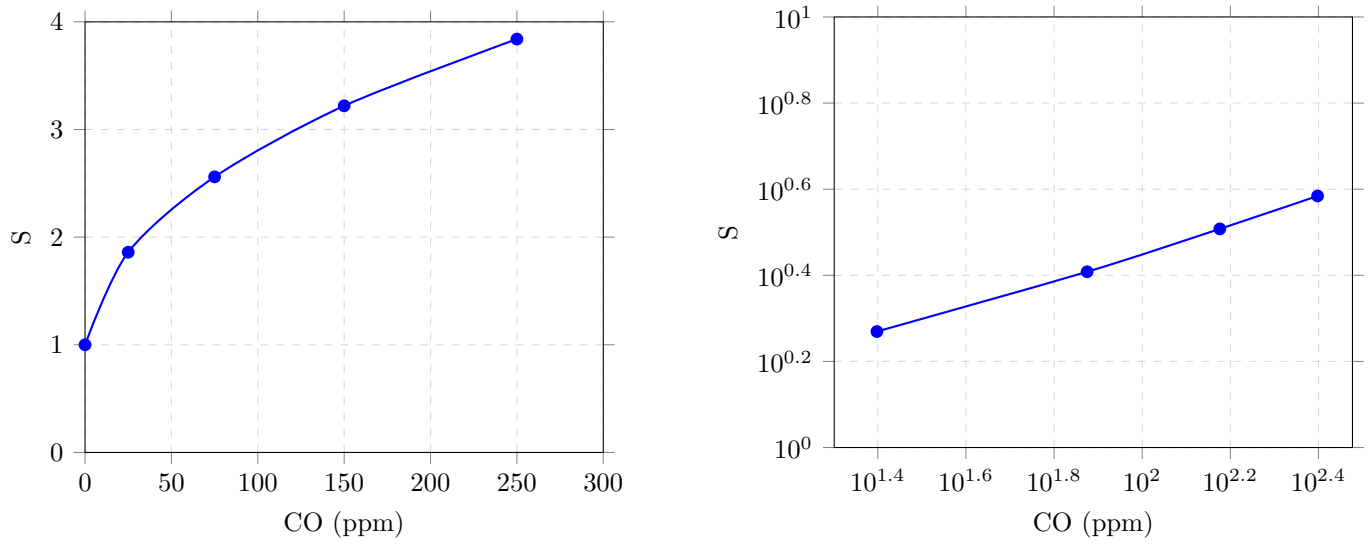
Due to the omnipresence of oxygen in most applications and the generally higher sensitivity with depletion layer controlled sensing mechanisms [40], n-type materials make up the biggest share of SMOX materials on the gas sensor market. Today, SnO_2 is among the most widely used base materials besides WO_3 and In_2O_3 [45]. It can be shown that the relationship between a sensor's resistance R and the partial pressure p_{CO} of CO , i. e. concentration, can be described by an allometric function [46].

$$R \propto p_{CO}^{n} \tag{2.1}$$

This power law calibration curve of SMOX gas sensors [47] can be be linearized in a double logarithmic presentation [48]. The log-transform therefore presents a frequent data treatment [17]. The exponential factor n in equation 2.1 is closely related to the sensitivity and sometimes interpreted or compared directly. An exemplary model calibration curve for an n-type SMOX gas sensor is given in Figure 2.2 for CO.

Typical definitions of the sensor signal S for SMOX gas sensors are given in equation 2.1 and put the resistance R in the presence of a stimulus in relation to the resistance in its absence R_0. The definitions of the baseline resistance R_0 may very much depend on the application and can factor in different reference states like relative humidity background. This treatment partially compensates different baseline resistances. The sensor signal is defined in a way that $S > 1$, that is n- and p-type sensor materials and reducing and oxidizing target gases have inverse relations, respectively [39]. Other definitions exist as well. For instance the raw resistance is useful for it's simplicity without the need of a baseline knowledge. Relative resistance changes express a sensor signal that starts zero and convey a large sensitivity when given in percent. As a consequence, analytical sensitivity – the sensitivity weighted by the standard deviation at a given concentration – is a useful measure to truthfully compare and rank different sensors, especially when yielding different sensor signal units due to different measurement principles [49].

$$
\begin{aligned}
&\text{n-type} \\
S &= \begin{cases} \frac{R_0}{R}, & \text{reducing gas} \\ \frac{R}{R_0}, & \text{oxidizing gas} \end{cases} \\
&\text{p-type} \\
S &= \begin{cases} \frac{R}{R_0}, & \text{reducing gas} \\ \frac{R_0}{R}, & \text{oxidizing gas} \end{cases}
\end{aligned}
\tag{2.2}
$$



(a) (b)

Figure 2.2: Exemplary calibration curve for CO.
(a) linear
(b) double logarithmic

2.2 Sensor Arrays & Temperature Modulation

Missing specificity or selectivity is the main drawback of SMOX gas sensors. Much effort is spend on synthesizing and tailoring new materials, but two other often more successful approaches are presented here. The first relies on the combination of different materials with (partially) orthogonal properties crossing out the effect of interfering gas specious. This practice produces a set of sensors called a sensor array with a multitude of outputs/readings that jointly are more robust [14, 50]. Though originally not considered to be a tool to tune selectivity or sensitivity [51], today a lot of attention is paid to the operation temperature of SMOX gas sensors. Besides the holy grail of a room temperature sensor with low power demands [52], the heater temperature is often optimized to favor certain surface reactions. Ruhland et al. for instance report a change from oxidizing effect to a reducing one in the case of NO_2 sensing with SnO_2 at higher temperatures.

Furthermore, the operation temperature may be modulated dynamically not only to reduce the power footprint via duty cycling [53], but to yield different specific responses and features, useful in dealing with mixtures of gases [54] or targeting other shortcomings like selectivity or stability. Still, rapidly changing temperatures may exert stress on the sensor, reducing its lifetime [55]. A good review about temperature modulation is given in by Lee and Reedy [18] who recognize that „[...] the true relationship between the conductance of a semiconductor sensor in the presence of reducing gases and the temperature of the sensor surface is very complex“. Many processes and traits depend on the temperature; ranging from the charge carrier concentration due to thermal activation, over the adsorption and desorption rate of oxygen species to the activated surface reactions giving rise to the „desired“ sensor signal. The most common modulation strategies involve the cyclic modulation with sinusoidal heater voltages [19, 56–58] or rectangular pulses.
Disregarding surface chemistry, (doped) SnO_2 exhibits thermistor like properties with Negative Thermal Coefficient (NTC) characteristics [59]. Schultealbert et al. [60] could show that temperature modulation could reduce or suppress sensor drift. They describe the opposing long and short-term effects in a temperature cycle, modeling the adsorption and desorption rate of ionosorbed oxygen species at the SMOX grain surface. For relatively fast changes, the non-equilibrium state depends on the history; e. g. an increased amount of ionosorbed oxygen is still present when abruptly decreasing the temperature. The relaxation of transient states into an equilibrium or steady state then depends on the ambient gas composition and concentration. The resulting conductance changes were very reproducible over a longer period especially for high resting temperatures around 300 °C. The exact nature of the surface oxygen species is still subject of active debate and research. Sanjines and Lévy report a maximum oxygen ionosorption capacity of SnO_2 at 640 K ($\approx$ 365 °C), in the upper range of typical SMOX gas sensor operation temperatures. These findings like many others are underpinned by Arrhenius type plots where the resistance is plotted against the inverse of the absolute temperature.
The two sensor types most commonly used in this work are the SGP30 [61] – which has reached its End Of Life (EOL) – and its successor SGP40 [62]. Both sensors are fully integrated MEMS type sensors in an (SMD) package with four distinct sensing materials a. k. a pixels and dedicated Application-Specific Integrated Circuit (ASIC) for temperature control and resistance readout. The sensors are digital and expose an Inter-Integrated Circuit (I^2C) interface. A single sensor unit already implements four sensing elements and materials, representing a sensor array. From a consumer point of view, the main difference are the individually tunable hotplates in the later SGP40 for each pixel/sensing element. While the exact nature of employed sensing materials is a well kept trade secret, initial variants were designed around SnO_2 doped with different concentrations of Pd [63]. Evaluating the resistance response, it is safe to argue, that n-type materials made it into production. It is important to state, that the results presented here are based on an unofficial sensor Application Programming Interface (API), differing from the recommended operation conditions. The multivariate nature of the sensor array's readings or those of a complex temperature profile necessitate advanced multivariate algorithms in contrast to simple univariate calibration curves.

2.3 Chemometrics

Chemometrics is a discipline combining methods from various fields like multivariate statistics, computer science, and analytical chemistry to address the task of extracting chemical information from multivariate data. Here, multivariate refers to multiple observed variables and multiple properties to be explained or predicted. Essentially, algorithms, methods or models are developed to allow a mapping from one set of explanatory variables $\mathbf{x}$ to another set of response variables $\mathbf{y}$, represented as vectors. For more than one observation, that is for a population or sample stemming from an experiment, these vectors can be concatenated to yield the matrices $\mathbf{X}$ and $\mathbf{Y}$, respectively. Depending on the context and field, many equivalent names are used for them [64]. Common terms are collected in Table 2.1. Most dependent variables are of numeric nature, but classes or attributes called labels could be used as well. Machine Learning (ML) models can be loosely grouped by the nature of independent and dependent variables. Methods that don't use labels or targets are unsupervised. Typical examples are the Principal Componenet Analysis (PCA) or the hierarchical cluster analysis aiming to group the observation into classes. A PCA has been used for instance to find and interpret the difference in light activated SMOX gas sensors in the sensing of CO and NO_2 [65]. In contrast, supervised methods seek to explain or predict these classes or numerical targets. Regressors are models with a continuous predicted variable distribution, whereas classification gives discrete outputs.
Most machine learning and chemometric problems are ill-posed and as such don't have one exact solution. Instead, the models are optimized to describe or predict a data set as well as possible. Typically, some model parameters are optimized to minimize an error like Mean Relative Error (MRE) and Mean Square Error (MSE) or to maximize an accuracy metric, such as the coefficient of determination R^2. Different algorithms for this mathematical optimization exist and are well established in many fields, where analytical solutions do not exist or are not computationally possible, e. g. in quantum chemical geometry optimizations. These optimization methods are typically iterative, improving a solution or approximating a local extremum step-wise. Common choices in ML tasks to find local minima are different flavors of a Stochastic Gradient Descent (SGD) [66]. Similar to the Newton-Raphson method, where the slope or gradient is used to iteratively estimate a function's root, the gradient is calculated from a randomly down-sampled (thus stochastic) set of data points. One particularly often used extension is Adaptive Momentum Estimation (Adam) [67], where the learning rate is adapted for each iteration, reducing the risk to oscillate around a local extremum without reaching it. In training ML models like CNNs, it is common to define a number of iterations a-priori instead of surpassing a given accuracy threshold or resolution. This approach is called „early stopping " and is a form of regularization that aims to reduce overfitting [68].
This work focuses mainly on regressors, trained on SMOX gas sensors' resistance readings to predict continuous concentrations of target gases. Most chemometric models rely on linear algebra or work best for explaining (multi-)linear correlation. It is well established to calibrate SMOX gas sensors using allometric functions or a power-law (equation 2.3), so the calibration curve can be linearized using a double logarithmic representation [69].

$$y = k \cdot x^a \tag{2.3}$$

If not stated otherwise, this approach is followed in pre-treating the data with the natural logarithm of the resistance ($\ln R$) and the natural logarithm of one plus the target gas concentrations ($\ln(1+c)$). The latter is needed to deal with concentrations of 0. All metrics like the MRE and visualizations in figures are based on the real target domain; that is after training a model on logarithmized data the predicted values are treated with the exponential function minus one ($\exp(x) - 1 \equiv$ `expm1`) to yield the correct concentration unit.

Table 2.1: Often encountered equivalent terms in machine learning, chemometrics, and multivariate statistics.

x	**y**
independent	dependent
input	output
regressor	regressand
predictor	predicted
explanatory	explained
feature	label or target

Hyperparameter Tuning

As briefly touched upon in the previous paragraphs, supervised chemometric models have to be trained or optimized for an analytical task at hand. Similar to fitting a function by the least squares method, a number of known $\mathbf{X}$ and $\mathbf{y}$ pairs, labeled data, is used to minimize an error of prediction or maximize precision. In this optimization a set of model parameters is found, like the slope or offset in (multi-)linear regression. This process is called training. Especially with growing complexity and therefor amount of trainable parameters, a sufficient training data set is needed. Models with many parameters trained on few examples tend to overfit and generalize poorly. This means, they „learn“ to predict the targets of the test set perfectly or very well, yet fail to explain outputs based on inputs that have not been in the test set with decent accuracy. To overcome these limitations many strategies exist and might strongly depend on the method being used. In general, to assess the generalization ability of a model, a complete labeled data set is split into a train and a test set. As before, the model is trained, that is parameters are optimized, on the train set. However, the model is evaluated – in terms of accuracy, Limit of Detection (LOD), or other metrics – not on the same samples, but on the reserved test set.

Not all aspects or parameters of a model can be trained. Untrainable parameters, like the number or connectivity of neurons in a Neural Network (NN) or the metric that has been optimized, are called hyper parameters. Searching the best hyper parameter(s) is referred to as hyper parameter tuning and typically requires another subset of the whole data set. Models differing in their hyper parameters are optimized on the train set, the best of them is pointed out by the performance in a validation set and the final precision is obtained in the test set [70].

Ridge Regression

Ridge regression is a regularized form of the Ordinary Least Squares (OLS) method for Multiple Linear Regression (MLR), estimating the coefficients for highly correlated or collinear input variables. While this type of regularization is also called Tikhonov regularization after Andrey Tikhonov, the term ridge refers to the shape of the response surface of the functions optimized [71]. The hyperparameter α – sometimes λ due to it being a a Lagrange parameter – penalizes the size of the coefficients by their L_2 norm. This way, no single input or feature is appointed too much importance. The same procedure based on the L_1 norm is called a Least Absolute Shrinkage and Selection Operator (LASSO) regression and the combination of both is referred to as elastic net [72]. These hyperparameters are also found in other applications like in the regularization in CNNs (see section 2.3). Table 2.2 lists the choices of α tuned in the ridge regressions in this work, implemented in the scikit-learn library [73].

Table 2.2: Hyper parameters of the used ridge regressors

Hyperparameter	Values
α	0, 1×10^{-3}, 1×10^{-2}, 1×10^{-1}, 1, 1×10^{1}

XGBoost

XGBoost classifiers and regression models, that have dominated many machine learning competitions, have the same architecture as random forests of decision trees [74]. However, they differ largely in the way they are trained or optimized. Depending on chosen or optimized hyperparameters, a certain number of these Classification And Regression Trees (CARTs) is taken as an ensemble yielding an aggregate of a prediction. This way a regression can take place and produce continuous outputs. Each tree recursively splits the data set based on selected features and „grows“ until a maximum depth is reached. In prediction, a tree is traversed along branches according to binary criteria until a terminal node leaf is reached. This node leaf is the prediction of the individual CART. `Subsampling` between 0 and 1 refers to a random fraction of the training samples used in growing each individual tree. It is a mean to avoid overfitting. The choices for hyperparameters used for XGBoost regressors in this work are listed in Table 2.3.

Table 2.3: Hyper parameters of the used XGBoost models

Hyperparameter	Values
Depth	6, 12, 24
Subsampling	1, 0.8, 0.6
Number	100, 200, 400

Limit of Detection & Analytical Sensitivity

Univariate, linear models are especially useful for a descriptive definition of multiple metrics and concepts. Figure 2.3 shows a calibration curve, that expresses the measured output y, e. g. resistance, of a sensor or analytic instrument in dependence of a concentration x, e. g. in ppb. The distribution functions for two exemplary concentrations point out that the measurand is subject to noise. The calibration function is often obtained from a set of (repeatedly) measured known concentrations. To receive the detected concentration for any measured output, the inverse of the calibration function, the analysis function is used. It is important to note, that calibration may be used to interpolate for a set of discrete known concentrations, but is expected to fail for extrapolation. The LOD for a given analyte, instrument, and process is defined – e. g. by the International Union of Pure and Applied Chemistry (IUPAC) [75] – as the lowest concentration that can be differentiated from a blank sample with a given statistical significance. The most straightforward method of determination is based on a regular calibration curve and evaluation of multiple blank samples. The distribution of measured outputs for blank samples around the mean $\hat{y}$ yields a critical value y_c, above which a sample is sufficiently different from a blank – i. e. a content has been detected – with the probability of α for a type I error. This critical values can be expressed in the concentration domain by the analysis function producing the LOD.

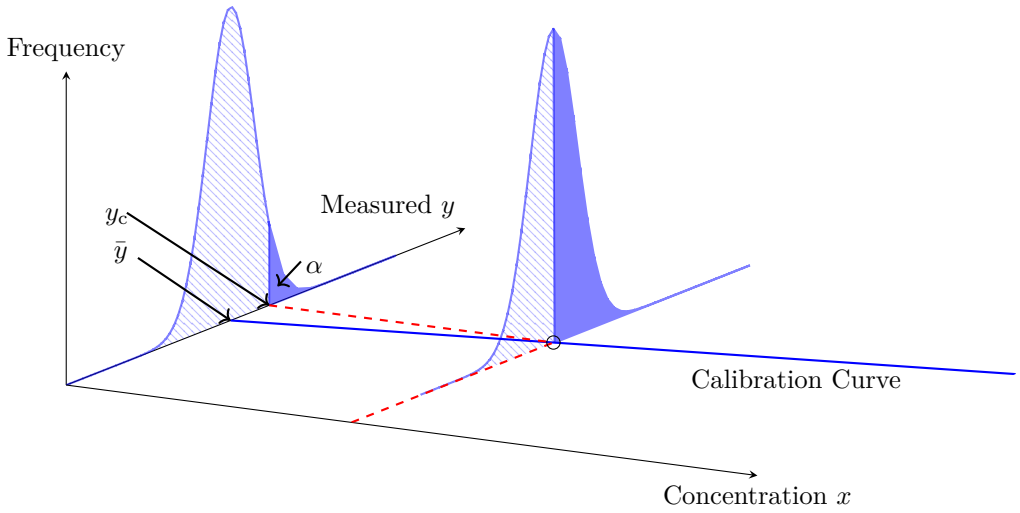


Figure 2.3: Visualization of the LOD definition according to DIN32645 [76].

$$\hat{c}_i = \hat{b}_0 + \mathbf{r}_i \cdot \hat{\mathbf{b}} \tag{2.4}$$

$$\mathbf{r}_{\mathrm{LOD}} = \mathbf{r}_{\mathrm{blank}} + 3 \cdot \mathbf{s}_{\mathrm{blank}} \tag{2.5}$$

$$s_{\mathrm{blank,i}} = \sqrt{\sum_j \frac{\left(r_{j,i} - \overline{r}_{\mathrm{blank},i}\right)^2}{n-1}} \tag{2.6}$$

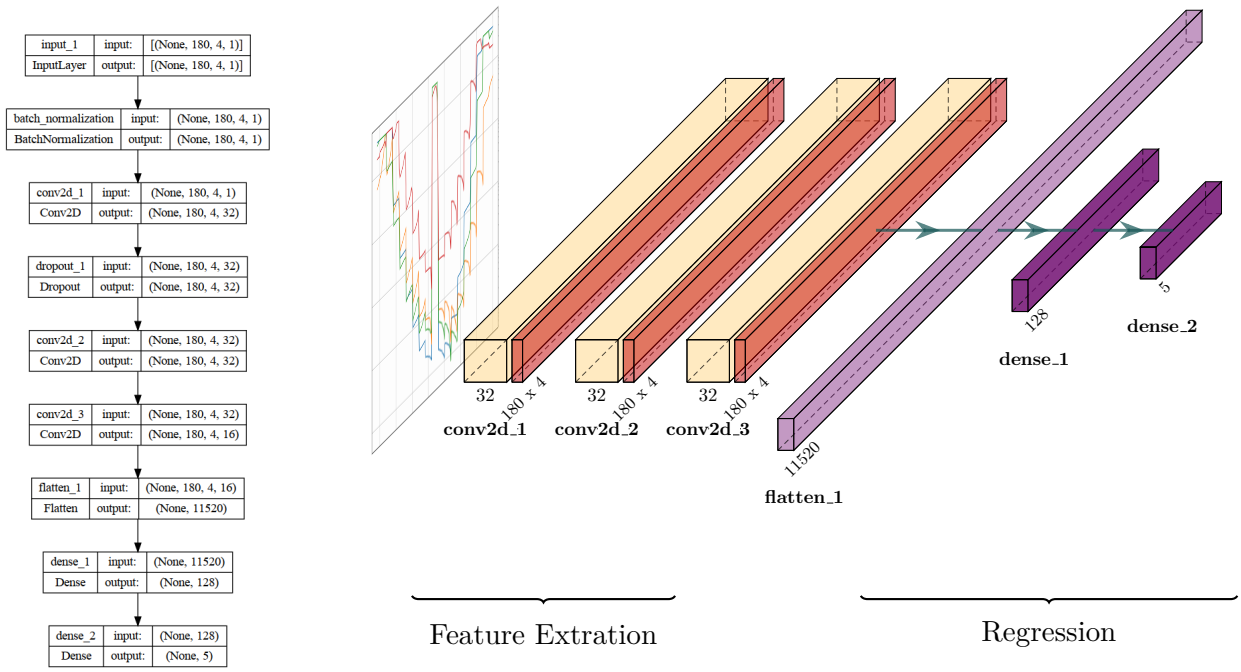
$$\tag{2.7}$$

This univariate definition of the LOD can be extended to multivariate regressors [77]. When the estimated or predicted concentration $\hat{c}$ is expressed by a (multi)linear analysis function in equation 2.4 using a vector of resistance features $\mathbf{r}$ as inputs and fitted parameters for the gradient $\hat{\mathbf{b}}$ and offset $\hat{b_0}$, the critical value in the input dimension $\mathbf{r}_{\mathrm{LOD}}$ (not concentration domain) is given by equation 2.5. The measured resistance(s) when exposed to a sample is considered to be significantly different from that/those of a „blank“ sample without the stimulus present if the resistance(s) deviate(s) from its mean resistance by more than three standard deviations[1]. Without an offset, the standard deviation of the blank can be approximated by the standard error of the gradient. The latter approach is especially useful when blank samples are not accessible – as is the case for reported atmospheric ozone concentration. Similar approaches are estimating the variance of a hypothetical blank sample based on the lowest encountered concentration or a regression thereof. The (United States) Environmental Protection Agency (EPA) describes different scenarios in their revised assessment of detection [78]. Further simplification is possible by defining the predicted concentration as the sensor signal. In this case a plot of predicted vs. true concentration with a linear fit function becomes a calibration curve.

[1] NB: When using raw resistances of an n-type SMOX with reducing gases, one needs to subtract three σ due to the negative slope.

Convolutional Neural Networks

An excellent overview about different ML models with an emphasis on deep learning is given in [79]. Originally developed and brought up in the field of CV, CNNs have quickly found other applications ranging from all kinds of time series data analysis [80, 81] to natural language processing [82] and natural sciences [83]. These models are a type of feed-forward neural network and accept tensors of the shape (n, x, y, c), where n is the number of observables processed at once, x and y are the height and width of the input, for instance the number of pixels in the columns and rows of an image. c is the number of channels in the input, e.g. 3 for a (RGB) image. CNNs may include a number of different layer types; an exemplary CNN architecture that has been used in this work is displayed in Figure 2.4. The name defining convolutional layers or blocks consist of a number of trainable filters or kernels of a given two-dimensional size and differentiate CNNs from simpler fully connected Deep Neural Networks (DNNs). In the feed-forward process they are convolved across the input with a given step size and produce two-dimensional feature maps by the dot-product of the filters and the input. During the training phase, these filters are optimized to extract certain features. In image recognition, the early convolutional blocks are are well thought of as finding outlines like edges or corners, whereas the following blocks can extract more complex features. Due to the convolution, CNNs are spatial invariant and can for instance find objects in images independent of their actual position. In the case of gas sensor responses, shifting baselines can be handled well. The size of images, inputs or feature maps can be greatly down sampled by pooling layers, that process partitions, e.g. by taking the maximum or average. In the end, the feature maps are typically processed by layers that are fully connected and resemble the architecture of a regular multi-layer perceptron. The finally obtained output is a vector that can describe the numerical targets in a regression task. For classification, the final vector is often one-hot encoded, i.e. it's a vector with n bits – the number of possible classes – where the single high bit corresponds to the predicted class. All CNNs in this work are implemented using `tensorflow` [84]. The generalized architecture of the models investigated is shown in Figure 2.4. Subfigure (a) is a simplified flow chart representation produced by `tensorflow`. Subfigure (b) is a graphical depiction of the same model emphasizing size relations and containing more information, like the number of filters per convolutional block or the number of neurons in the fully connected layers. The code to produce the latter image is heavily inspired by [85]. Batch Normalization is an optional layer, that normalizes the input features in a batch – the number of samples used in the iterative training after which parameters are updated – and allows for higher learning rates and faster convergence [86, 87]. The L_2 regularization applied to the kernel (weights), bias and activity of the convolutional blocks aims to reduce overfitting similar to the motivation in the ridge regression. Another strategy to improve generalization can be introduced by dropout; the process of randomly dropping or nullifying features during training with the intent to create more robust networks, that can deal with suboptimal or noisy data. Lastly, the size of the kernels, the number of filters and neurons is optimized.



(a) Flowchart (b) Spatial representation

Figure 2.4: Typical architecture of CNNs used in this work.

Limit of Detection & Analytical Sensitivity

While the LOD and analytical sensitivity for ANNs and CNNs could be derived from internal weights and parameters using simulated noise or statistics [88], a more straightforward and simplified approach uses the predicted target concentration as a sensors signal. The predicted vs. true concentration plots can be treated as the calibration and be fitted with linear calibration functions. Associated uncertainties and confidential intervals can be calculated.

Transfer Learning

Essentially all chemometric models or even simple calibration curves require the consideration of sensor to sensor variation and repeatability. CNN models may be used to mitigate sensor to sensor variation [89]. Typically, these algorithms with a large number of trainable parameters are highly specialized when trained on single sensor units (or arrays). Classically, some sensors are (re)calibrated on wafer level, batches or single units with a reduced data set and the required time is crucial in production [90]. A recalibration scheme for CNN is transfer learning, where pre-trained models are used instead of a random initialization of weights and other parameters before the optimization. Thereby fewer (re)training samples are needed, reducing the (re)calibration time [91].

2.4 SHAP Analysis

The metric Shapley value named in honor of Lloyd Shapley can be interpreted for predictive models as the contribution of individual input features and is formally defined in equation 2.8. Here, $\varphi_i(v)$ is the Shapley value of player i for a characteristic function v describing a game. N is the set of all n players. The Shapley value is calculated as the contribution of a single player by comparing the payout of coalitions of players S that are including that player with those that are not. Averaging happens over all permutations of coalitions. Shapley values are additive, i. e. the sum of all Shapley values equals the payout.

$$\varphi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(n - |S| - 1)!}{n!} (v(S \cup \{i\}) - v(S)) \tag{2.8}$$

In chemometric problems, features take the role of players and the function v is best described by a model giving a prediction (or an accuracy or error thereof) and is therefore dependent on the number of features n used in the training. An exemplary calculation of the Shapley value for the R^2 score of the ozone prediction by a ridge regression and a temperature modulated SGP40 sensor is given in 2.4. To average over all weighted combinations of features, all permutations of the feature set are noted down, as in the first column. Next, the property of interest is calculated for S and $S \cup \{i\}$; the sets of features excluding and including the evaluated feature, respectively. The set S is derived as all features that appear in order before the feature of interest. The difference between those two values is then noted for each permutation. Lastly, those differences are averaged to yield the marginal contribution of said feature to the overall measurable. Here, the coefficient of determination R^2 is used to describe how well the model can explain and predict variances in the data. Equation 2.9 defines R^2 as one minus the ratio of residual squares SS_{residual} to the total squares SS_{total}. While R^2 may be smaller than zero and larger than one numerically, it is typically considered to be normalized or confined to boundaries zero and one, where one is a perfect prediction and zero corresponds to a model not able to explain the variance in the data at all. The calculation can be simplified, if the model under investigation is agnostic of the order of features, i. e. $R^2(i,j) = R^2(j,i)$ as is valid for the ridge regression.

$$R^2 = 1 - \frac{SS_{\text{residual}}}{SS_{\text{tot}}} = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (y_i - \overline{y})^2} \tag{2.9}$$

Table 2.4: Exemplary procedure of calculating Shapley values for the R^2 score for a ridge regression predicting ozone with a temperature modulated SGP40 sensor.

Permutation	Features wo/ Px4			Features w/ Px4			Contribution of Px4 to R^2
1,2,3,4	R^2(1,2,3)	=	0.749	R^2(1,2,3,4)	=	0.761	0.012
1,2,4,3	R^2(1,2)	=	0.722	R^2(1,2,4)	=	0.773	0.051
⋮	⋮			⋮			⋮
Average							0.245

When the number of features n increases, this thorough calculation becomes unfeasible as the permutations scale with $n!$. For the four pixel SGP gas sensors with 180 resistance readings in series, $n! = 720! \approx 1.8 \times 10^{1742}$ is astronomically large and beyond grasp. Lundberg et al. [32] have adapted this concept of Shapleys and extended it with approximations to various machine learning problems to evaluate feature importance. These SHAP values are model agnostic and can be used to quantitatively compare different methods or algorithms that become explainable instead of being a hard to interpret black box as is otherwise the case for most very complex models.

3 Material and Methods

This section introduces the experimental setups. The used and developed programs, methods and devices are presented. Especially the frameworks `esp32ota` (see section 3.2.1) and `PyGUI` (see section 3.1) are adopted and used widely in the work group, the author is a part of. They are thus described in detail and may serve as additional documentation. `PyMFC` has been rolled out in the whole laboratory end of 2022, replacing the former software of controlling tens of Gas Mixing Systems (GMSs). The hardware around the `esp32ota` framework including the reference sensor units of the GMSs is in use by the hundreds.
Similarly, the versatile eNose Platform platform developed with *AO Action* is highlighted in section 3.2.3.

3.1 Gas Mixing System

When creating gaseous mixtures or samples under laboratory condition, GMSs were used for autonomous and continuous operation. Both, static and dynamic conditions can be generated with a GMS. For the more common dynamic operation conditions, typically a constant total gas flow is achieved by dosing the appropriate amounts/flow rates of analyte gases to be diluted and making up the rest with a carrier gas, often synthetic air. The flow rates are regulated by Mass Flow Controllers (MFCs) with additional magnetic valves, that ensure an air tight system with channels that could be turned off completely. Relative Humidity (RH) is adjusted by humidifiers, where the sufficiently large head space is saturated with water vapor at room temperature. A typical GMS setup is shown in 3.1. The RH, ambient temperature and general gas composition are monitored with a downstream reference sensor module comprised of a HYT939 (digital temperature and humidity sensor, IST, Switzerland) and a SMOX sensor. Previously, a TGS2600 (Figaro Engineering, Japan) has been in use and has now been replaced by an SGP40 (Sensirion AG, Switzerland) sensor array. The same hardware as in the `esp32ota` framework (see section 3.2.1 is used with simplified micropython code operating the SGP40 at 300 °C.

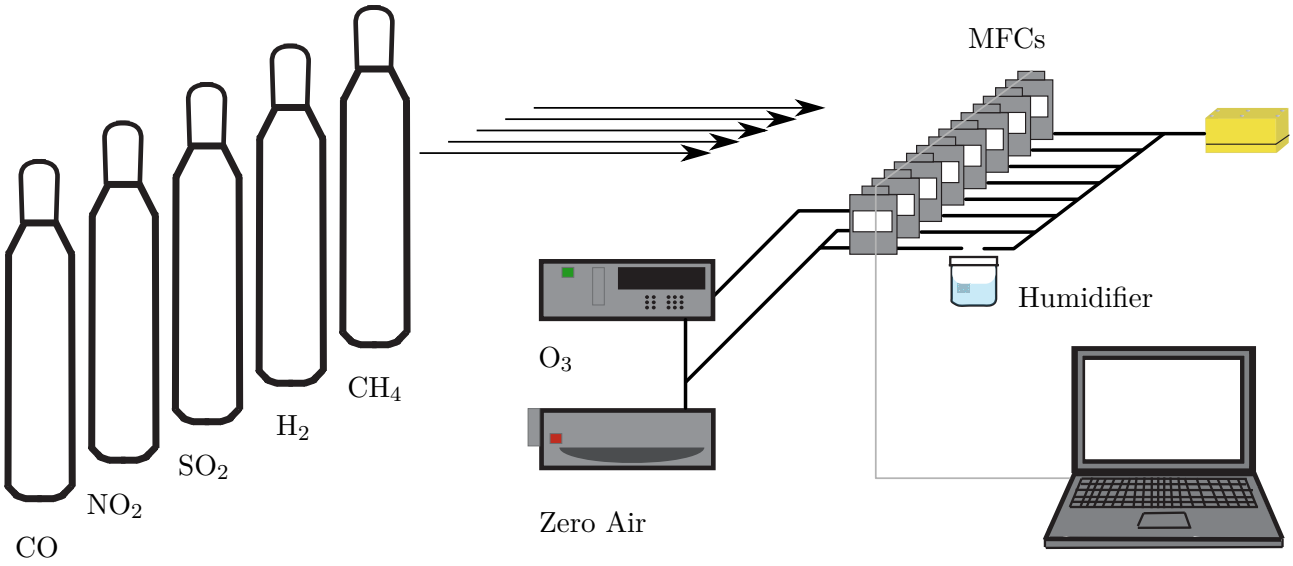


Figure 3.1: Gas Mixing System

A custom Graphical User Interface (GUI) has been developed in this work to autotomize the GMS operation and experiments. While many features like database connectivity, test dialogues, and new references sensor units have been implemented by the author, a solid foundation in Keysight VEE (Visual Engineering Environment) existed before and had been maintained by Peter Bonanati. The latest software is written in python using PyQt5 and implements an abstract base class `PyGUI` that realizes the general GUI and methods to read input files and save data in output files as well as data bases. It is inherited by `PyMFC` and a number of similar GUIs sharing the most basic functionality. The family of custom measurement software is depicted in Figure 6.1 in the Appendix (see page VII). The common base class implements many methods to create the graphical interface and sets up a timer with a callback for continuous operation. The exact behavior has to be overridden by the child classes. `QbPy` is a program to control a static GMS, for instance in a box with a volume of one cubic meter (thus „Qb“). It has been developed in by Luca Völkl in the frame of his Master thesis [92]. As gases are dosed via MFCs, a lot of the functionality is inherited from `PyMFC`. The major difference lies in the static nature of the setup which essentially aims to mimic environments closer to typical applications. The gas concentration cannot be changed quasi-instantaneously as in the dynamic GMS, where the concentration is controlled by diluting a flow of target gas with a carrier gas to maintain a constant total flow. Instead, the injected total volume is summed up by integration of the flow over time and put in relation to the total volume of a measurement chamber. The third main program is `PyScanner` which offers a tool to control measurement equipment like multimeters or multimeter scanners (hence the name). A lot of the development in terms of hardware control has been driven by Benjamin Junker. Each of the child GUIs is tested with a `PyTest` class which checks for the most important functionality, namely if a measurement can be started and if the program can be updated. These unit tests are executed by gitlab [93] runners on an in-house server to minimize the risk of rolling out a faulty version.A screenshot during the operation is shown in 3.2. The upper left box contains the controls

to start and stop the measurement, a numeric index, and the time of the current condition applied. The stats table (b) below summarizes the state of all the MFCs of the GMS with set and actual flow rate and resulting concentration. The plugin table (c) lists commands and responses transceived from external equipment. The log text box (d) logs the output of the program with a given loglevel which is also saved in a temporary file. Status bar (e) contains the current base and application version and quickly shows by color if the system is connected to the intranet for storage. The evolution of MFC openings is displayed as a time series plot in (f) and is complemented by the simultaneous readings of reference sensors (g).

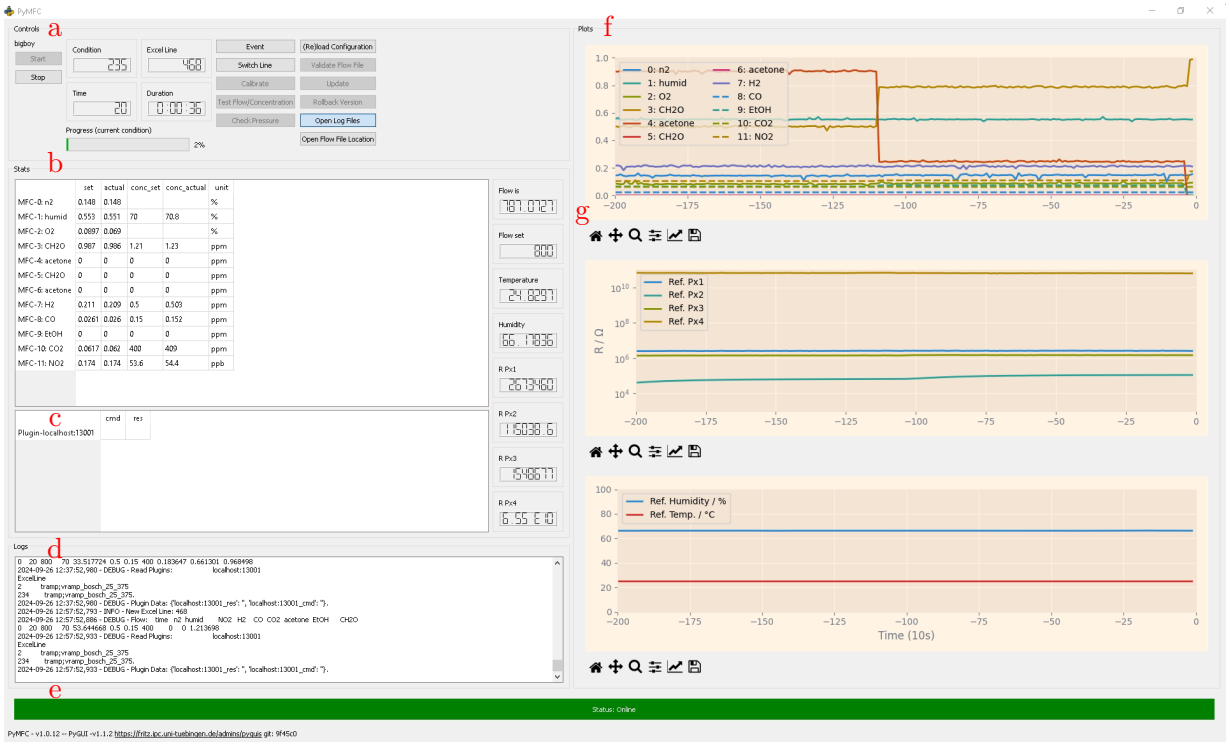


Figure 3.2: Gas Mixing System
(a) Controls and line count
(b) Stats table with data for each gas channel
(c) Plugin data
(d) Logs
(e) Status bar and footer with version numbers git hash
(f) Time series plot of channel openings
(g) Time series plots of reference sensor data

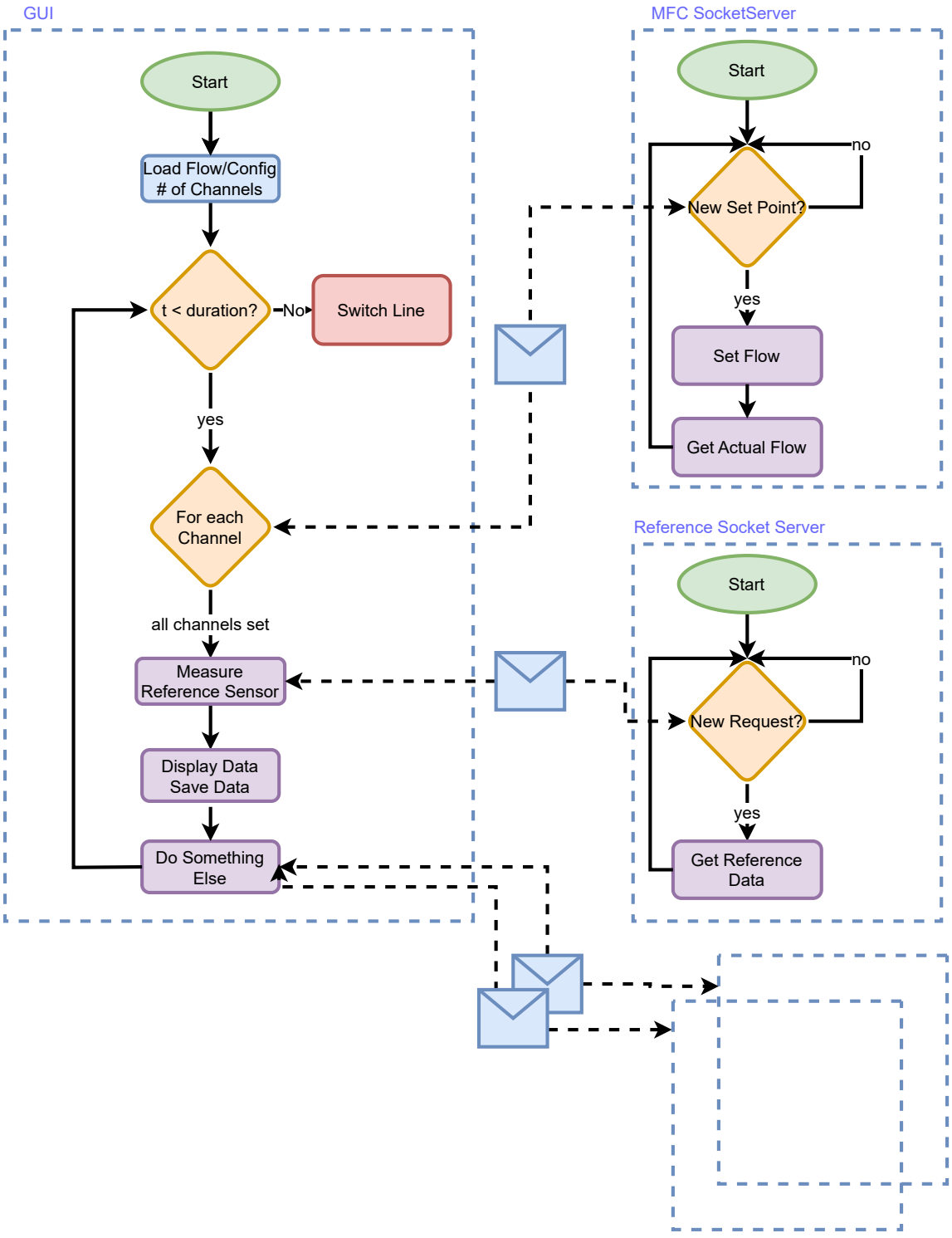


Figure 3.3: GMS/MFC Software Flowchart

Figure 3.3 contains flow charts, illustrating different parts of the software. Each dotted box describes a single process, taking care of a defined task, which in turn may also be multi threading. The most heavy-lifting process is the GUI process itself, which does not only create the interface but is also the brain of the operation. After an input file is loaded, which describes the desired experiment, used GMS and time series of gas conditions, the control loop starts. Every few seconds (10 s by default) the program evaluates if a new gas condition needs to be created, which requires calculation of instrument setpoints (e. g. of the MFCs, corresponding to flows). In that case, the newly determined setpoints are fed into the other program processes via network sockets. The socket servers accepting these messages are typically run on the same computer/host and have been started automatically, but may also run on different systems for maximal flexibility. For different types of hardware, individual socket servers have been developed. After the appropriate hardware has been accessed, a reading or actual value may be returned, which is recorded by the main process. All current setpoints as well as the actual hardware readings and the desired and resulting gas concentrations are recorded in the stats table. This is useful for monitoring the proper operation of all hardware parts and quickly identify malfunctions, like empty gas bottles resulting in too low of a flow. Large deviations are highlighted by a channel's red row in the table. Errors like this are handled accordingly with emails sent to the user. Another piece of useful hardware are the reference sensors that are operated down stream. Their readings are recorded and displayed in the GUI on the two lower panels on the right hand side. The obtained data are recorded in local text files (European Data Format (EDF) and Comma Separated Values (CSV)) and optionally uploaded to an in-house database for continuous observation. The status of the upload is highlighted by a status bar on the bottom that stays green for successful uploads and otherwise red. Overall, the GUI is designed to be very flexible and work with many different GMS setups, may it be a different number of used or connected gas lines or the combination of altering hardware types. Additionally, custom commands may be sent to custom socket servers to allow the easy integration and control of other hardware, for instance regulating the ambient or sensor temperature and illumination [65]. This communication is optionally triggered with a new gas condition as defined by the excel input file and recorded in the „plugins“table.
The PyGUI repository is maintained with a single branch for releases of all the children classes. The releases' tags are of the form `<class>-v<major>.<minor>.<hotfix>` where `<class>` is the GUI class affected and the version numbering scheme follows the semantic versioning [94]. Figure 3.4 shows a segment of the git flow's releases. Each release could bundle a number of commits or merges not displayed. The different classes and their releases are highlighted by color, the parent base class `PyGUI` is blue. The chosen segment starts with three consecutive hotfix releases of PyMFC, followed by a feature implemented in the parent class in release `PyGUI-v1.1.3`. One of those PyMFC releases turned out buggy or problematic and has been removed along with its tag. Lastly, PyScanner get's a new relase. The middle and right column illustrate exemplarily to which version/release different GUI versions would update. `PyMFC-v1.0.8` will update to the latest parent class release `PyGUI-v1.1.3`. `PyScanner-v1.1.0` will update to the `PyScanner-v1.1.2`. For this behaviour, the software tracks and compares the parent and child class' release versions. This way it is ensured, that the latest yet relevant commits and releases are pulled.

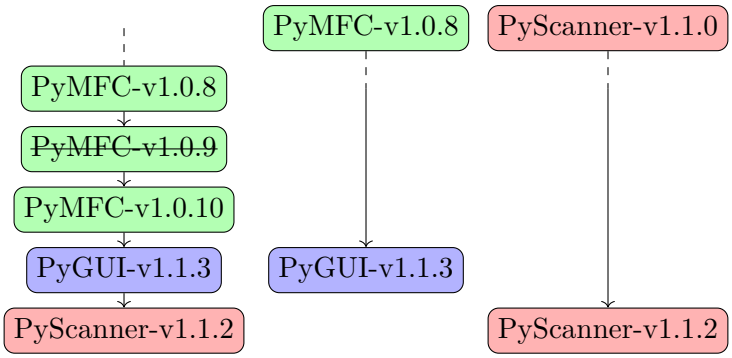


Figure 3.4: Exemplary segment of the git flow with PyGUI releases.
The two columns on the right illustrate update behaviour.

3.2 Sensor Operation

Most commercially available, digital sensors use a well established and defined data bus for control and read out. Often encountered protocols are Universal Asynchronous Receiver / Transmitter (UART), Serial Peripheral Interface (SPI), and I^2C. All digital gas sensors used in this context, as well as those implemented in the analog sensor shield (see 3.2.2) use the I^2C bus, which uses one controller and possibly multiple targets, i.e. the sensors. Each target has a unique address, that is either fixed or can be shifted by a few bits to allow multiple units of the same type connected to just on bus. Besides two wires for the supply voltage, a serial clock and data line are needed to connect I^2C devices: Serial DAta (SDA) and Serial CLock (SCL). In this work, two frameworks have been developed and used to control digital sensors, most importantly SGP30 and SGP40 gas sensors. The first is part of the analog sensor shield attached to a Raspberry Pi (RPi), the Sensor Raspberry Pi (SenPi), and exposes four I^2C buses behind an expander connected to the RPi (see Figure 3.13b). In contrast to this versatile system taking care of many tasks at once, small wireless sensor nodes were developed around Espressif's ESP32 microcontroller with up to two I^2C ports. A circuit board with an additional micro (SD) card for offline uses is shown in Figure 3.7. The software running on those boards is written in micropython, a small and efficient implementation of python for embedded systems and microcontrollers. The software framework is named `esp32-ota` as it is intended for the use with these boards and can receive updates via (WiFi), over-the-air (ota). The latter feature relies heavily on the pre-existing update functionality implemented by Ronald Dehuysser [95]. The different software versions are managed by gitlab releases. As described by the programmatic flow chart in Figure 3.9, the sensor nodes typically start by connecting to a WiFi and check for an ota update. Next, they will load their configuration from a central JavaScript Object Notation (json) storage on a central server. Lastly, they scan for multiple connected sensors, load the required libraries and instantiate sensor objects. Finally, they will control the sensors, reading them out with a default sampling rate of 1 Hz and applying the temperature cycle specified by the `tramp` argument. The argument `_id` matches the configuration with a given unit by the MAC address and `logdb` specifies the database to push data to. A gilgamesh instance offers an integration of a postgresql data base with an http API [96]. If not specified otherwise, the temperature ramp `25-375-dip` as described by listing 2 is used as an internal standard modulation.

The reference sensor modules each GMS is equipped with, see 3.1, are controlled by the same hardware. In this instance, the ESP32 running dedicated micropython code waits for the serial message `{"func_name": "get_data"}` before outputting the SGP40's resistance at 300 °C together with the HYT939's relative humidity and temperature values. The eNose Platform platform developed by and with AO Action implements a similar architecture and expects equivalent serial messages executing different functions, see section sec:enose. Another closely related, battery powered sensor unit with an SGP40, SD card, and Real Time Clock (RTC) is scheduled for a launch to the International Space Station (ISS) in March, 2025 to get a handful of daily readings in micro gravity. The findings are expected to be helpful in the continuation of developing an early fire warning system before ignition [97].

3.2.1 Digital Sensors

A single esp32ota module can control up to two I^2C busses with multiple different sensors. The Printed Circuit Board (PCB) is depicted in Figure 3.7. Here, it is not equipped with an ESP32, so the underlying micro-SD card slot can be seen. This storage solution offers the possibility to realize offline solutions outside the esp32ota framework, e.g. as a stripped down eNose Platform module. The module may be powered via the ESP32's (USB) port or via the pin header exposing 5 V and GND. Sensor modules may be conveniently connected with 10-pin flat ribbon cables and boxed headers/sockets. In the boxed header's default position `J4`, two pinouts of the first I^2C bus are realized, matching Adafruit's pinout and that of the internal Institute of Physical Chemistry (of the University of Tübingen) (IPC) SGP PCB. The former corresponds to the odd pin numbers on the front and the latter to the even pin numbers on the back side of the socket. This way, either sensor's PCB's 1-rowed pin strip may be directly inserted. If the boxed header is soldered to populate footprint `J8` with an offset, two I^2C busses using the internal SGP layout are exposed. The different positions and pinouts are laid out in Figure 3.5. The General Purpose Input/Outputs (GPIOs) as well as other parameters have to be defined in a global configuration to enable the flexible use of different ESP dev kits with differing pinouts. An exemplary configuration is shown in Listing Table 1. `_id` is derived from the ESP's MAC address for clear and robust identification in the form `hwid_MAC>`. In the `i2c` dictionary, the definition of the second I^2C's pins is optional. The sensors' data is saved in an in-house gilgamesh postgresql database `logdb` in table `name`. If possible, the sensors will be modulated according to `tramp`. The connected sensors and their types don't need to be specified, as they are dynamically scanned and initialized at the module's boot up. All supported digital sensors are listed in 3.1. Sensors that cannot be temperature controlled (as it often does not make sense, e. g. temperature sensor) are not affected by a selected temperature profile. Yet, a temperature profile is needed for programmatic reasons, but could contain a single measurement point or a large number thereof. The configuration of any esp32ota unit can be changed and adjusted via a web service exposing the the respective in-house json storage. A screenshot of said website is depicted in Figure 3.6.

	Row 1	**Row 2**	**Row3**
Pin 1	NC	GND	GND
Pin 2	3V3	3V3	3V3
Pin 3	GND	SCL	SCL2
Pin 4	SCL	SDA	SDA2
Pin 5	SDA	GND	GND
	Adafruit I2C-1	IPC I2C-1	IPC I2C-2

Figure 3.5: Different pinouts of the 10-pin Insulation-Displacement Contact (IDC) header connected as default (dashed) or offset by one row (dotted).

```
{
    "_id": "hwid_308398ef42e0",
    "i2c": {
      "scl": 21,
      "sda": 22
      "scl2": 21,
      "sda2": 22
    },
    "logdb": "esp32",
    "name": "test",
    "tramp": "300_const"
}
```

Listing 1: esp32ota configuration json.

```
{
  "time_per_step": 1000,
  "tramp": [[25,5],[50, 10], [100,
    ↪  10], [150, 10], [200, 10],
    ↪  [250, 10], [300, 10], [350,
    ↪  10], [375, 10], [25, 10], [375,
    ↪  10], [350, 10], [300, 10],
    ↪  [250, 10], [200, 10], [150,
    ↪  10], [100, 10], [50,
    ↪  10],[25,5]]
}
```

Listing 2: Temperature Cycle `25-375-dip`.

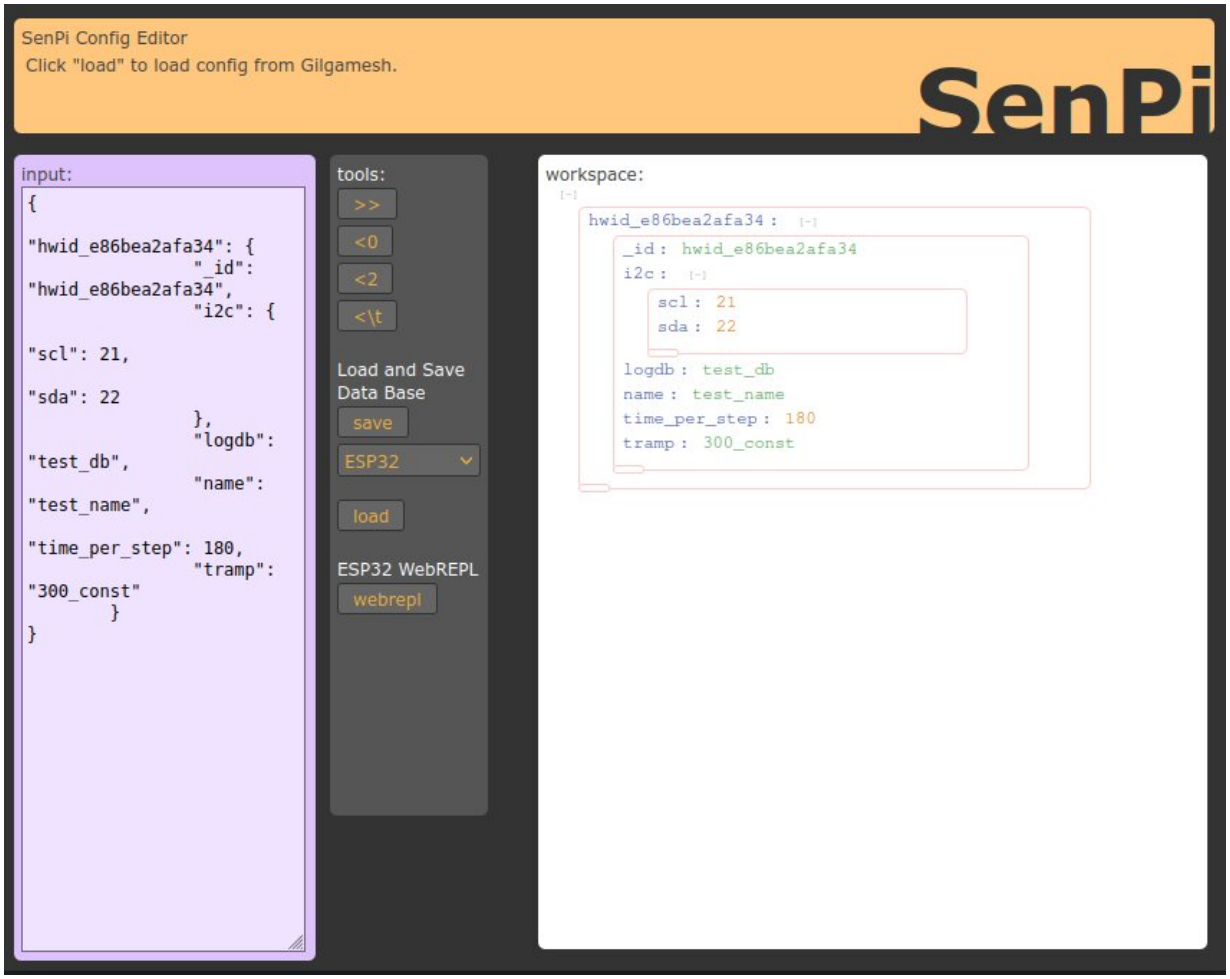


Figure 3.6: Web service to view and update the configuration of any known esp32ota module.

Since an esp32ota unit be powered by supplying 5 V to the pin header on the left, multiple devices could be powered in parallel from a single power source. For additional control or debugging – especially with issues of USB drivers – the UART port is broken out on the right at a 3.3 V logic level. This provides a reliable connection that could be leveraged by a Raspberry Pi without the use of USB.
A completely assembled esp32ota unit with an SGP40 sensor attached can be seen in Figure 3.8 next to a bunch of sensor PCBs. The sensor is housed in a custom Polyether ether ketone (PEEK) chamber with a volume of roughly 2–3 mL. Both, PEEK and Polytetrafluoroethylene (PTFE) chambers are well established in the field due to their chemical inertness. The former has the advantage of hardness which makes it easier to machine finer threads. The gas connection is established with Swagelok tube fittings with 1/8 " parallel (ISO) thread. The SGP40 PCB only exposes the sensor package on the top side without vias in the proximity. A gas tight seal is realized with an O-ring which can be seen in its gland on the right side of the image. The technical drawing in Figure 3.10 shows the exact dimensions and visualizes the duct and cavity of the sensor chamber. The reference sensors used in the operation of gas mixing systems (see section 3.1) in the work group are essentially equipped with the same module as shown in Figure 3.8 with an additional compartment for an HYT939 temperature and humidity sensor. Instead of publishing data automatically and wirelessly, they are programmed to wait for a serial command and only operate the SGP40 at 300 °C constant heater temperature.

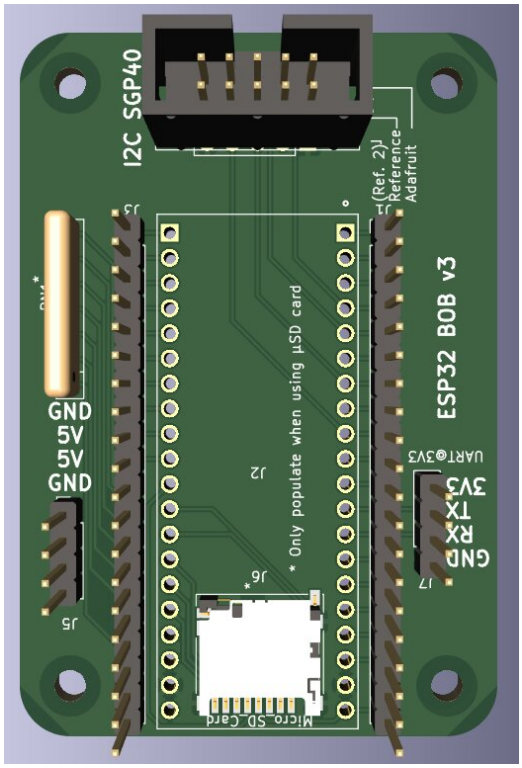


Figure 3.7: Partly populated esp32ota PCB
The boxed header exposes a single I^2C bus with two different pinouts on each side.

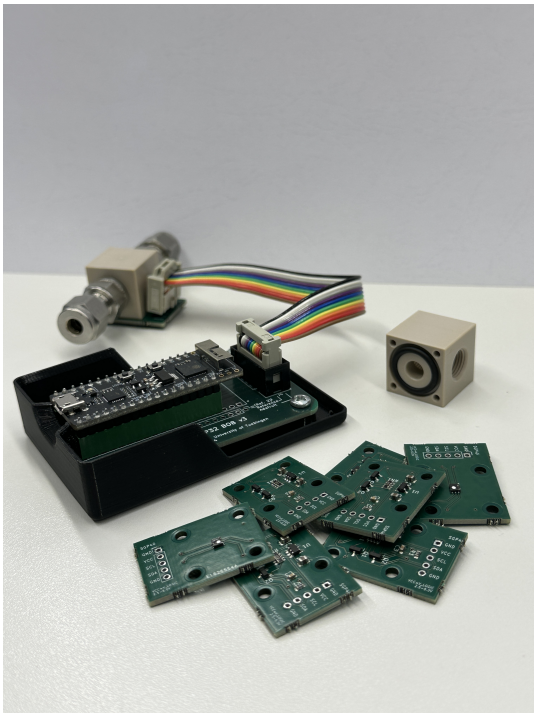


Figure 3.8: esp32ota module with SGP40 sensor chamber made from machined PEEK.

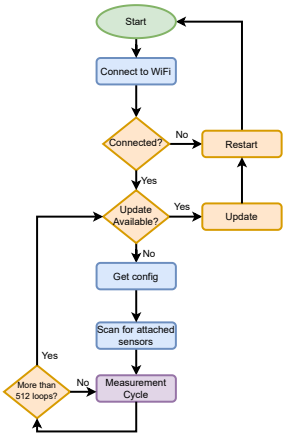


Figure 3.9: esp32ota software flowchart.

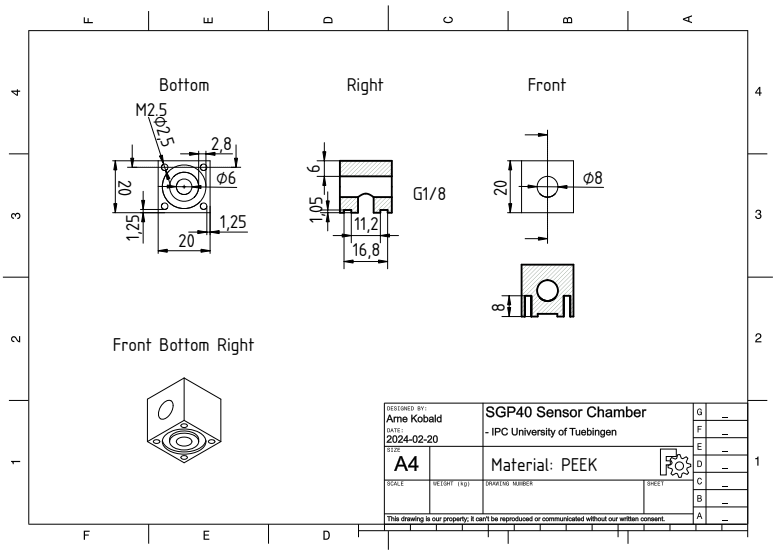


Figure 3.10: Technical drawing of the measurement chamber..

The program flow of the micropython firmware is illustrated in Figure 3.9. After starting – either by first power on or due to a soft or hard reset – the device connects to a (WLAN) if it's (SSID) and password are in the configuration. Next, it checks for new firmware on the in-house gitlab server. A potential update is applied resulting in a reboot. When the device configuration is received from the json storage, the I^2C bus is initialized and scanned for connected sensors of known type (as judged by their address) which in turn are instantiated and initialized. In the main event loop, the connected sensor(s) are operated in the defined modulation followed by a read-out and publishing of the data. This loop is only broken occasionally – every 512th cycle – to check for a firmware update. The sensor operation in the main loop is greatly streamlined with the use of sensors classes that share an abstract base class. Every sensor class has the `init_sensor` and `get_data` method. The latter returns a dictionary with the sensor data with keys in the form of `<sensor_type>_<measurand>`. A Watch Dog Timer (WDT) ensures proper continuous operation and resets the board whenever an error is encountered.

Figure 3.11 shows the relation of these sensor classes with the most important methods and attributes. Sensirion's SGP30 and SGP40 share another parent class implementing common methods like the conversion from two bytes (a word, 16 bit) to floats and Cyclic Redundancy Check (CRC). Only sensors that can be temperature controlled (`is_temp_tunable == True`) receive calls to the method `set_temperature` in the main loop. The SGP40 is special and allows individual hotplate temperature (`multi_hotplate == True`). Therefore the method `set_temperature` also accepts a list (or tuple) of length 4 to set all four hotplates instead of a single float. A complete list of supported sensors is compiled in Table 3.1. The majority are (MEMS) SMOX gas sensors joined by a number of gas sensors with different sensing principle. The CO_2 sensors leverage vibrational transitions and either utilize NonDispersive InfraRed (NDIR) adsorption (SCD30, CDM7160) or the photoacoustic effect (SCD4X) [98]. The SFA30 is an electrochemical cell for formaldehyde detection. The list of digital sensors is complemented by a group of temperature and relative humidity sensors. The ADS1115 is a Analog to Digital Converter (ADC) useful for measuring the voltage drop across an analog sensor in a voltage divider.

Table 3.1: I^2C sensors supported by esp32ota. Listed are the sensor type, manufacturer, measurands and if an operation temperature can be controlled.

Address	Sensor	Manufacturer	Measurands	Temperature Control
0x28	HYT939	IST AG, Switzerland	Temperature, Relative Humidity	✗
0x32	ZMOD44XX	Renesas Electronics Corporation, Japan	Resistance	✗
0x33	ZMOD45XX	Renesas Electronics Corporation, Japan	Resistance	✗
0x48 – 0x4B	ADS1115	Texas Instruments USA	Voltage, e.g. from Voltage Dividers	✗
0x52	ENS160	Sciosense B. v, The Netherlands	4 SMOX Resistances	✓
0x58	SGP30	Sensirion AG, Switzerland	4 SMOX Resistances	✓
0x59	SGP40	Sensirion AG, Switzerland	4 SMOX Resistances	✓
0x70	SHT30	Sensirion AG, Switzerland	Temperature Relative Humidity	
0x5d	SFA30	Sensirion AG, Switzerland	Formaldehyde Concentration	✗
0x61	SCD30	Sensirion AG, Switzerland	CO_2 Concentration	✗
0x62	SCD4X	Sensirion AG, Switzerland	CO_2 Concentration	✗
0x69	CDM7160	Figaro Engineering, Japan	CO_2 Concentration	✗
0x76	BME280	Bosch Sensortec, Germany	SMOX Resistance, Temperature, Relative Humidity, Pressure	✓
0x76/0x77	BME680/8	Bosch Sensortec, Germany	SMOX Resistance, Temperature, Relative Humidity, Pressure	✓

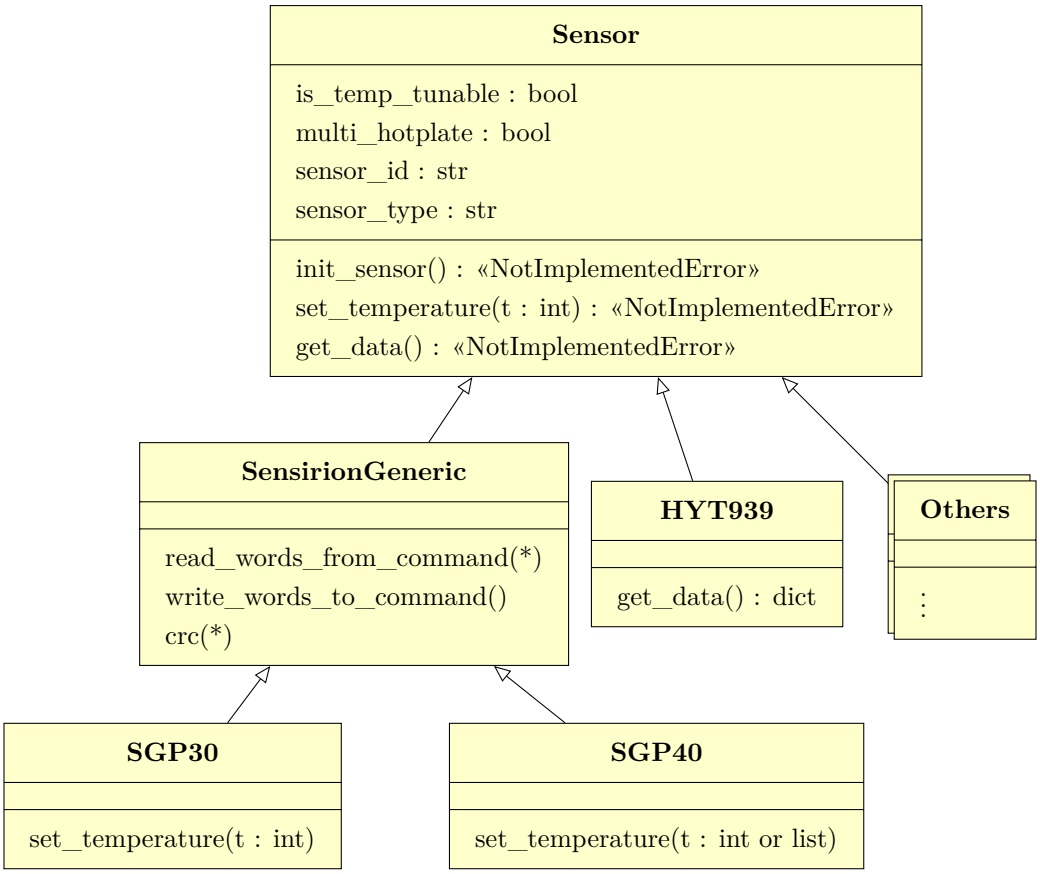


Figure 3.11: Family tree of the different sensor classes with selected attributes and simplified methods.

Temperature modulation modes can be defined and managed by the same web service as for the configuration. The internal standard temperature cycle is called `25_375_dip` and is defined in Listing 2. This temperature mode features 19 discrete temperature steps starting from 25 °C increasing to 375 °C, with a sudden temperature dip to room temperature, before decreasing again stepwise. This type of modulation is well established [99, 100]. It combines the idea of screening the space of temperature and thermal energy to enable distinct chemical surface reactions [56] with the concept of probing transient states induced by sudden and large temperature changes [60]. Leah Schynowski compared different modulation cycles of SGP40 sensors applied for the prediction of NO_2, H_2, CO and Toluene in a background of 5, 30, and 70 % RH in her Masters thesis [101]. The cycles investigated possessed the same shape as the `25_375` with elevated minium temperatures or featured pulses of room temperature between each temperature step. The resulting features did not produce grossly different accuracies while the overall space of temperature covered ranged from room temperature or slightly warmer to well above 350 °C. Unsurprisingly, all modes outperformed a constantly heated counterpart.

The typical `time_per_step` in milliseconds is 1,000 and is the interval between two measurements (in fact their starting times for finite acquisition durations). Consequently, the corresponding sampling frequency is 1 Hz. This is a sufficiently long period to establish the desired operation temperature. With the implementation here, an SGP30 could in theory be operated with a frequency close to 3 Hz. Commands that change the temperature take longer, which is another reason for the chosen sampling frequency. The τ_{90} response time is well below 0.5 s for both heat-up and cool-down processes between 35 and 375 °C at ambient room temperature. The successor SGP40 could be operated with up to 5 Hz. The `tramp` is defined by a nested iterable, a list of pairs. Each pair determines the operating temperature and the number of measurements resulting in the time in multiples of `time_per_step`. After iterating through all of those pairs, a new temperature cycle in the same fashion is begun. In multivariate data analysis, the sequence of measurements – e. g. the resistance time series – can be considered a feature vector of a single object (sometimes called a single multidimensional data point). To label and identify each point throughout a single cycle as well as the individual cycles, the following control variables are introduced in `esp32ota`: `ramp_start` is the absolute time stamp in epoch format when a cycle was started and stays constant throughout it. `temp_step` starts newly at 0 each time a new cycle begins and increases by one with each temperature change. `ramp_step` also starts at 0, but increases with every measurement point. To highlight these interrelations, Figure 3.12 illustrated the evolution of these variable through the course of two complete measurement cycles. Most important for data processing and cleaning, `ramp_start` is used to group measurement points of a single ramp together and judge for a complete cycle by count, see section 3.3.

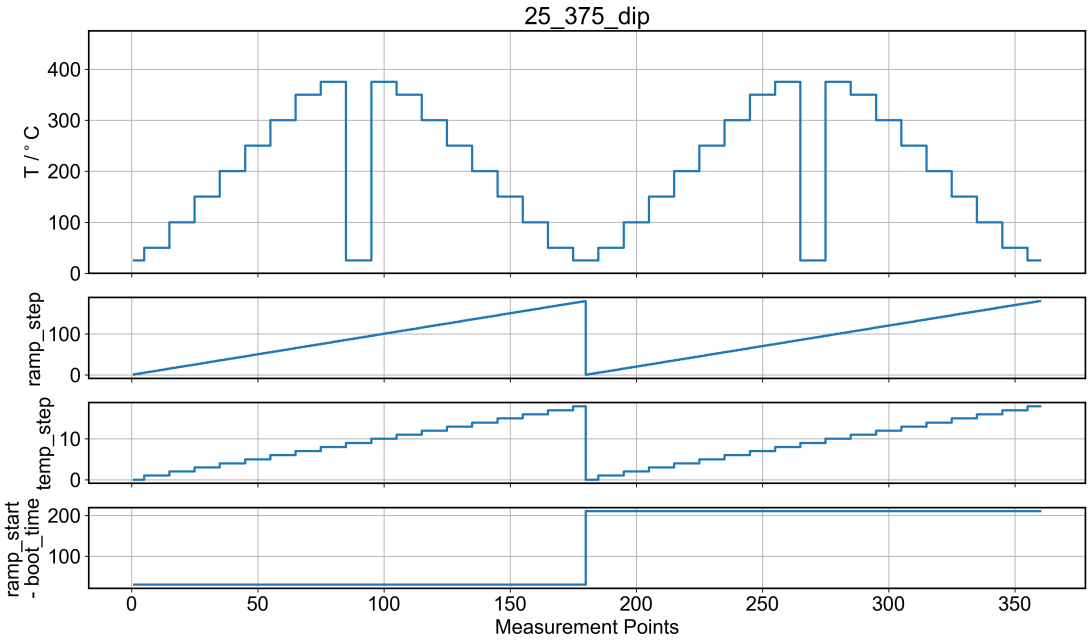
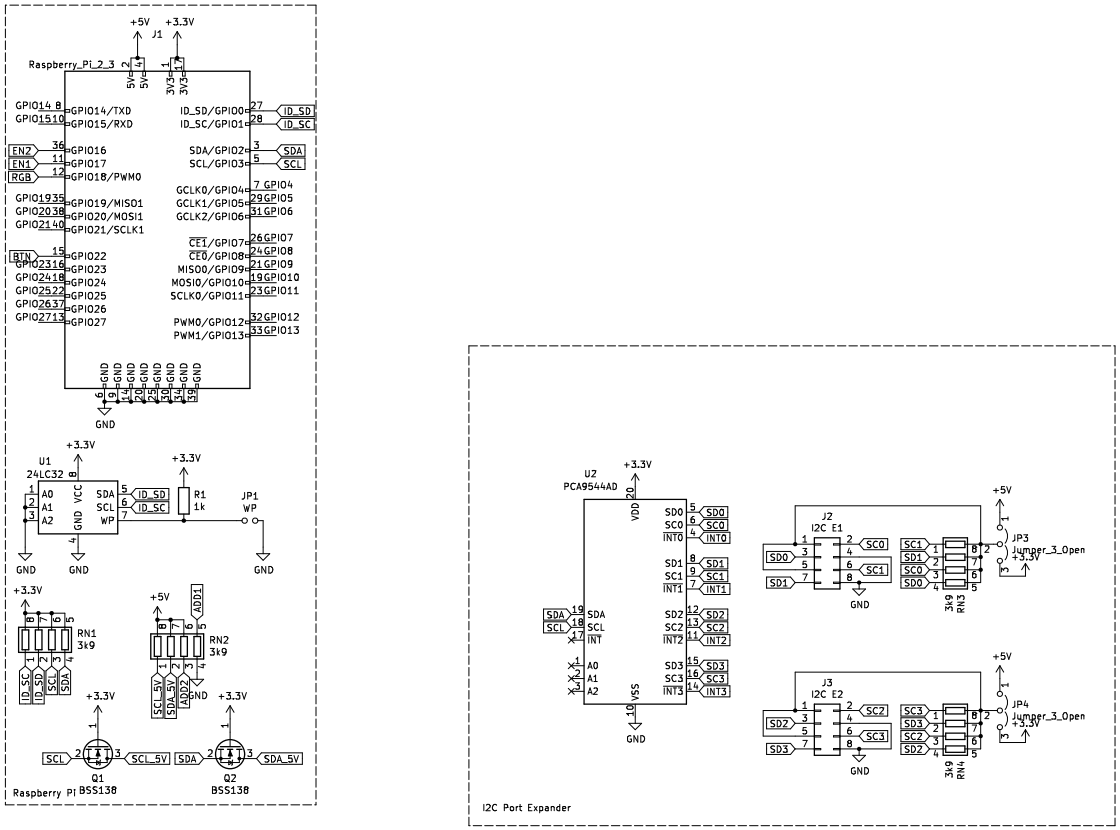


Figure 3.12: Time series of the operation temperature and associated control variables throughout two complete measurement cycles of the modulation 25_375_dip.

3.2.2 Analog Sensors

The SenPi Hardware Attached on Top (HAT) is an RPi attachment suitable for controlling and reading two analog SMOX gas sensors. More specifically, it can be used to set the sensors heater voltages and read the sensitive layers' resistances over multiple orders of magnitude, that can easily be encountered with temperature modulation or different (base) materials. Additionally, it features some convenient peripherals and an I^2C expander for attaching digital sensors. The SenPi HAT allows for the use of a stackable header, with the used GPIOs specified in Figure 3.13(a). The HAT is designed according to the Raspberry Pi Foundation's specifications [102] and as such features an Electrically Erasable Read Only Memory (EEPROM) that enables easy identification and the automatic loading of required kernel modules with a dedicated device tree overlay. Also pictured are the I^2C pull-up resistors as well as the level shifters for some chips requiring a 5 V supply. The I^2C port expander PCA9544 (NXP) [103], can be configured with jumpers to run on 3.3 V or 5 V-level as described in Figure 3.13(b). with the two 8-pin IDC sockets, up to four digital sensors with the same I^2C address can be connected.



(a) RPi header. (b) I^2C port expander.

Figure 3.13: SenPi connectors.

The analog sensor circuit – exemplarily depicted in Figure 3.14 for one of the two sensor ports – addresses two main tasks for operating an analog SMOX gas sensor: supplying a heater voltage to the heater (and optionally reading the resulting current for better control) and reading the actual resistance of the sensitive layer which is the output of the sensor. The circuit in Figure 3.14 (a) addresses the first task and revolves around the I^2C controlled buck-boost converter MP8862 ((MPS)) [104]. The readout relies on reading voltages with the ADS1115 (TI) [105]. For instance, the output heater voltage is read after scaling down with a voltage divider and the heater current is measured as a voltage drop across a 1 Ω shunt resistor as shown in Figure 3.14 (c). The resistance of the sensing layer on the other hand is measured with the LOG114 (TI) [106]. This logarithmic amplifier is designed to measure small currents over a wide dynamic range (8 decades, between 100 pA and 10 mA) in relation to a reference current. The circuit on the right side of Figure 3.14 (b) is based on TI's reference design. Here, the reference current is generated with a precision voltage reference of 4.096 V and a constant resistor of 1.62 MΩ. The reference current of around 1 mA (see equation 3.1) is four orders of magnitudes higher than the minimal current that can accurately be measured. In principal, this relates to maximal measurable resistances in the GΩ-range. The current through the sensitive layer is generated by the same voltage reference. Equation 3.2 describes the ideal transfer function of the LOG114 and relates the logarithmic voltage $V_{\log}$ with the ratio of reference and input resistance. In general, the current ratio is used in the transfer function, but can be substituted by the inverse ratio of resistances when the same voltage is applied across them. $V_{\log}$ is then scaled using the internal Operational Amplifier (OpAmp) as described in equation 3.3 before being measured by the ADC. The scaling factor S_{factor} is set by resistors R10 and R11 and the offset V_{offset} by resistors R5 and R6 (see Equation 3.4).

$$I_{\text{ref}} = \frac{4.096\,\text{V} - 2.5\,\text{V}}{1.62\,\text{M}\Omega} = 0.99\,\mu\text{A} \tag{3.1}$$

$$V_{\log} = 0.375 \cdot \log_{10}\left(\frac{R}{R_{\text{ref}}}\right) + V_{\text{ref}} \tag{3.2}$$

$$V_{\text{out}} = -S_{\text{factor}} \cdot V_{\log} + V_{\text{offset}} \tag{3.3}$$

$$V_{\text{offset}} = (1 + S_{\text{factor}}) \cdot 2.5\,\text{V} \cdot \frac{R_6}{R_5 + R_6} \tag{3.4}$$

All constants in Equations 3.1 through 3.3 are subject to calibration and fitting with the real components used.

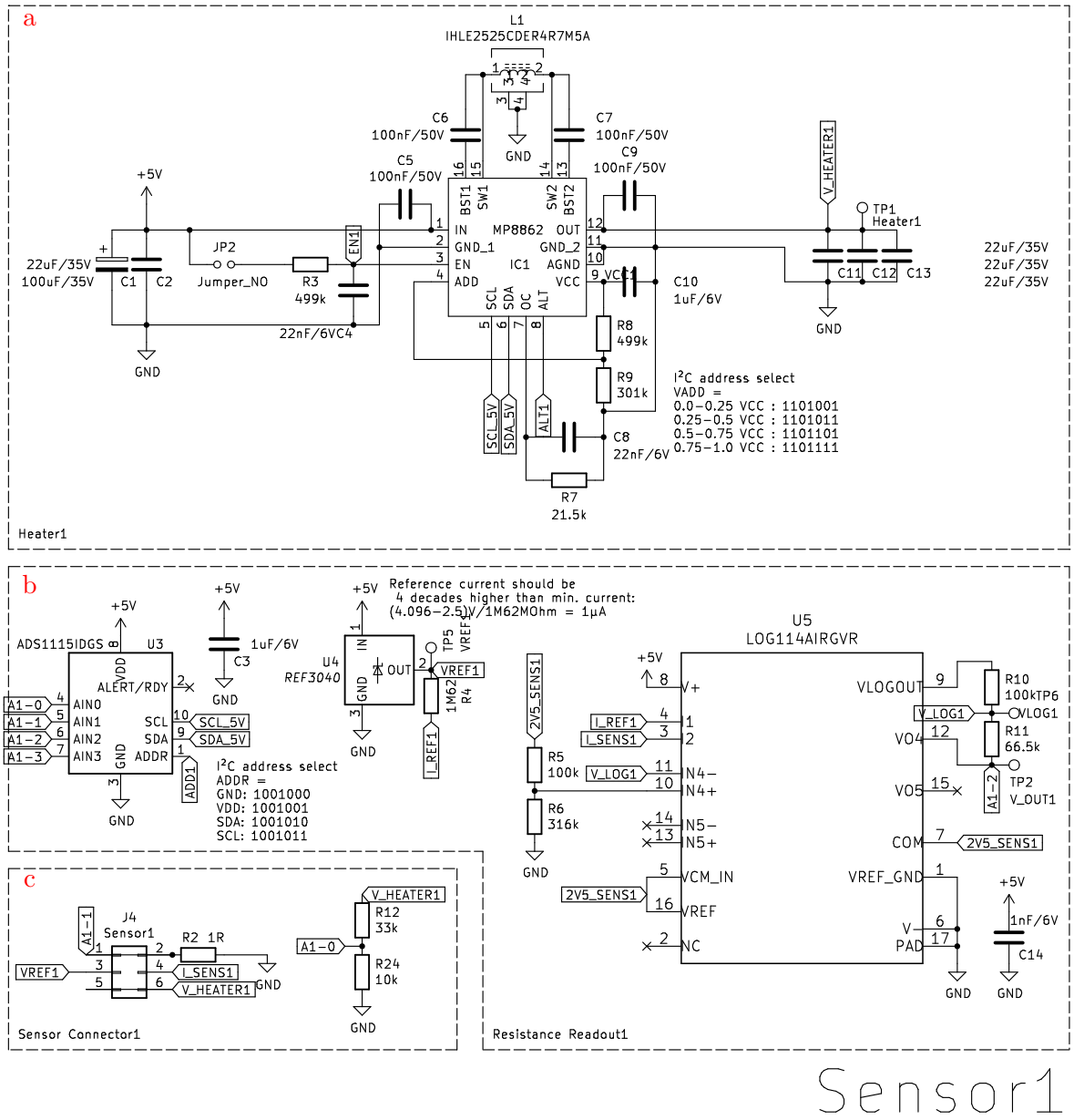


Figure 3.14: Analog sensor circuit.
(a) Heater circuit.
(b) ADC circuit for reading the transformed sensor resistance and heater voltage as well as heater current via a shunt resistor.
(c) Sensor IDC connector.

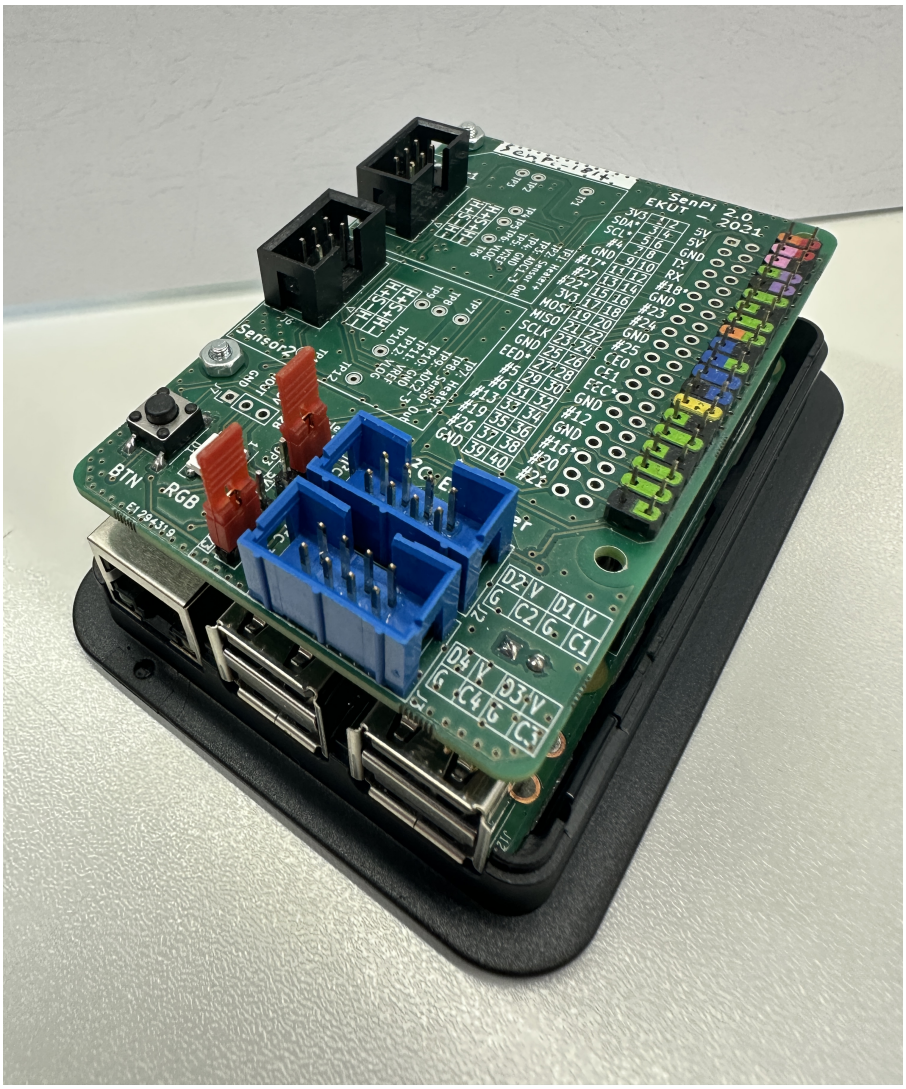
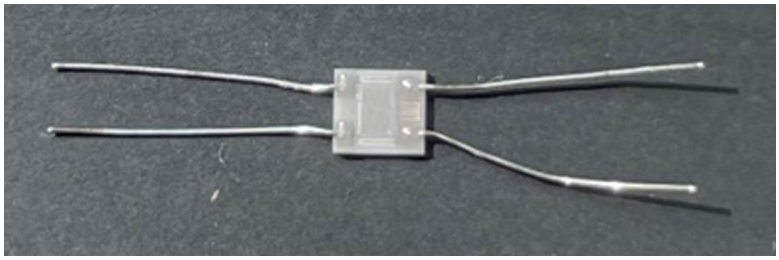


Figure 3.15: RPi equipped with a fully functional SenPi HAT.

A common housing for analog gas sensors is based on the Transistor Outline 12 (TO-12) standard [107]. A miniaturized Al_2O_3 ceramic substrate used for in-house deposition of sensing materials is shown in Figure 3.16(a). The Pt heater and electrodes have leads on the rim that can be spot welded to TO-12 bases as shown in Figure 3.16(b). While being miniaturized compared to many prototypes, the size and especially (thermal) mass of the substrate is still substantially larger than commercial MEMS sensors. Therefore, a temperature cycle with longer residence times is used.



(a)

(b)

Figure 3.16: Gas sensor on ceramic IST substrate with Pt heater and electrodes
(a) spot welded to a TO-12 base
(b). The PTFE measurement chamber is visible below.

In typical application of SMOX gas sensors, the operation temperature is controlled or determined by applying a certain voltage and measuring the heater resistance [108]. The calibration functions for an IST sensor as described above are given in Equation 3.6 for a voltage calibration and Equation 3.5 for a resistance calibration. The latter uses the the Positive Thermal Coefficient (PTC) properties of the Pt heater and is more robust against changes in the ambient temperature. Figure 3.17 show the voltage applied to the IST heater alongside the measured current and calculated resistance of the Pt heater. The temperature is obtained using the linear calibration curve, Equation 3.5. The typical operation temperature (cycle) has been raised by 50 °C – `75_425_dip` – in order to ensure the SenPi can supply the minimum voltage, calculated according to Equation 3.6. For comparison, Figure 3.18 shows the equivalent temperature cycle of an SGP30, which gives access to the actual heater temperature. The SGP's temperature changes almost instantaneously, whereas the IST shows significant lag, especially in the temperature dip.

$$R = 11.9\,\Omega\ + 0.0506 \cdot \frac{\vartheta}{^\circ\mathrm{C}}\,\Omega \tag{3.5}$$

$$U = 0.301\,\mathrm{V} + 0.0133 \cdot \frac{\vartheta}{^\circ\mathrm{C}}\,\mathrm{V} \tag{3.6}$$

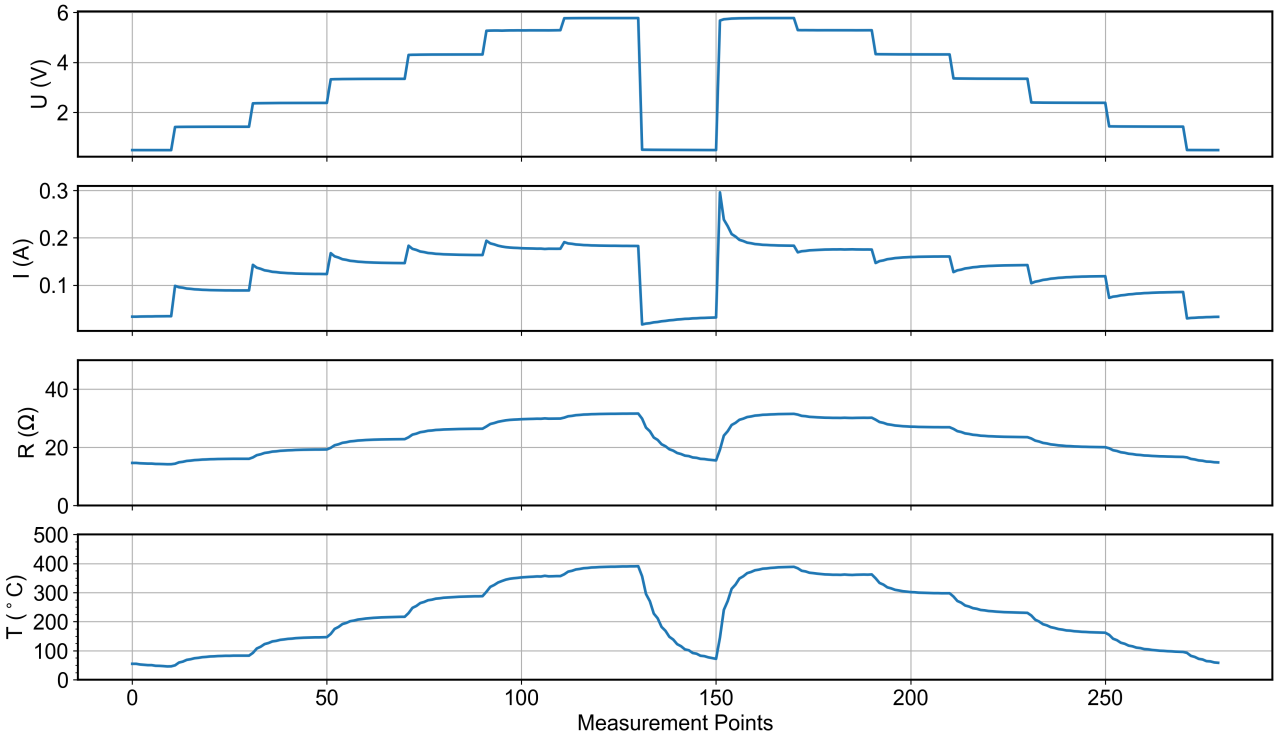


Figure 3.17: One voltage cycle according to 75_425_dip applied to an analog sensor with a SenPi.

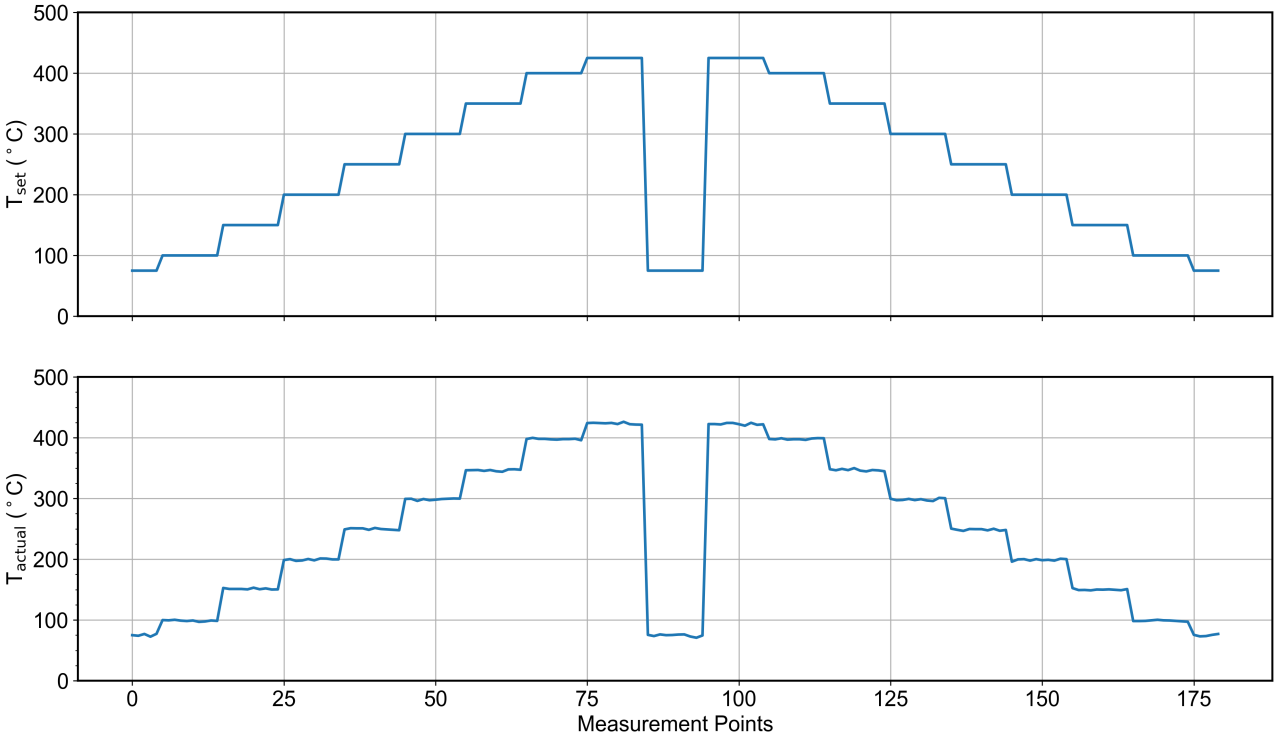


Figure 3.18: One temperature cycle 75_425_dip applied to an SGP30 that reports the actual temperature back.

3.2.3 eNose Platform

The eNose Platform platform described in this section has been provided courtesy of AO-Action, with whom it has been developed as a versatile sensing solution for many applications. It is designed around the microcontroller ESP32 and shares many building blocks in terms of hard- and software with the esp32ota project and the reference sensor units used in the IPC work group. The unique selling point is the device's autonomy to work as a standalone sampling and measuring unit, suitable for a wide range of application scenarios. It can measure static ambient air or draw in gaseous samples from up to four sources. In the default configuration, two micropumps deliver gas streams from samples under test or a reference, e. g. provided by filtered ambient air. Two commercial SGP40 sensor units can be controlled and read out, placed in PEEK chambers or open to the surrounding. However, the PCB is designed for flexible extension with broken out GPIOs. The unit's sampling mode can be configured and triggered with buttons on a front panel or via USB. The acquired data is logged on SD card, annotated with a timestamp from an RTC, in addition to the serial interface. The system can be employed in similar scenarios like competitive developments for sampling ambient air[109], in breath analysis[110], quality control like fish freshness[16], or process control [13].

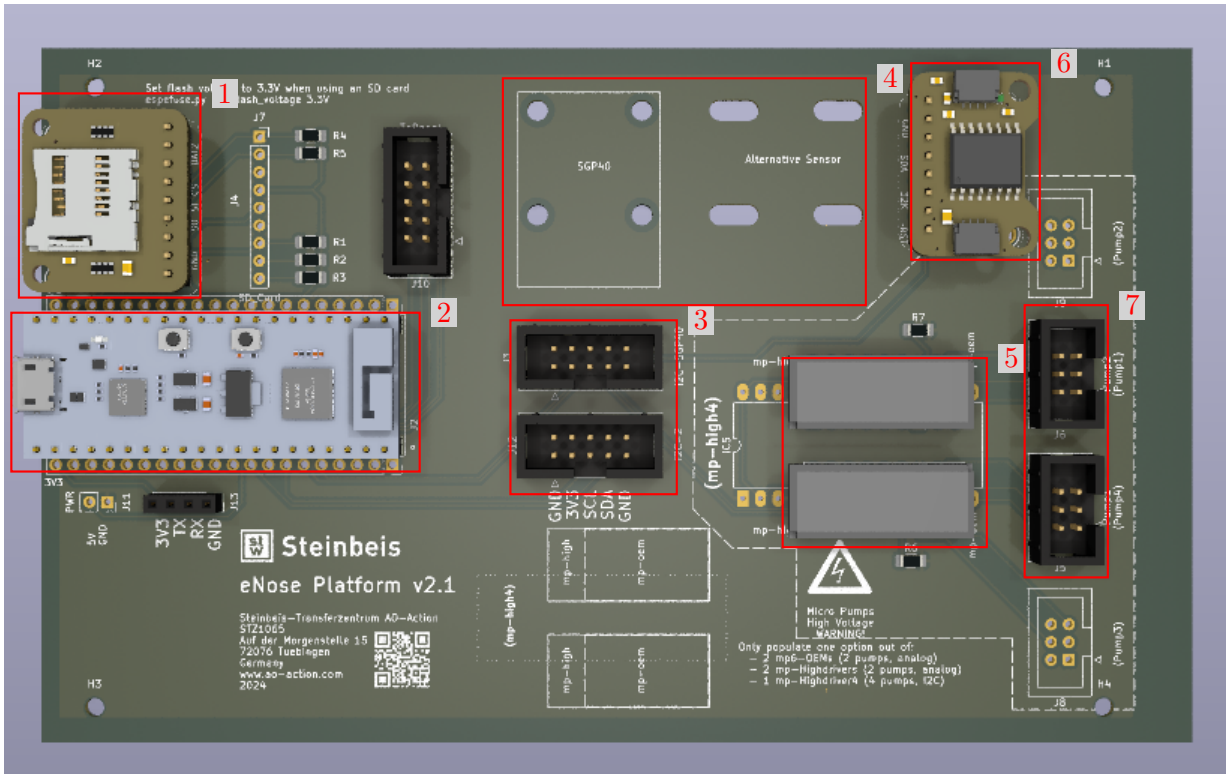


Figure 3.19: Populated main PCB of the eNose Platform platform.
(1) Micro SD card slot
(2) ESP32
(3) I^2C/sensor connectors
(4) Slots for SGP40s/sensor chambers
(5) Micropump drivers
(6) RTC
(7) Micropump connectors

The device's main PCB is displayed in Figure 3.19, with the ESP32 dev kit in rectangle 2. The system is equipped with a battery buffered RTC (rectangle 6) and micro SD (rectangle 1) card slot by Adafruit to enable offline measurements with accurate time stamps. The digital sensor typically used and mounted in a chamber that can be fastened to the board – rectangle 4 – is a temperature modulated SGP40 controlled via I^2C – rectangle 3. Other I^2C sensors may be added and implemented as well and a second bus allows for two sensors of each kind/address to be connected. Besides static measurements, where the sensors are exposed to the measurement medium or ambient air directly, dynamic operation is possible. Two Bartels micropumps driven by high voltage drivers – rectangle 5 – are connected – rectangle 7 – to actively draw gaseous samples from a source and reference. Sources could be gas streams as created by a GMS or sample bags. When needed or desired, this standard setup can be extended to four pumps and four independent gas sources.

```
{
  "btn1_func": {
    "func_name": "flush"
  },
  "btn2_func": {
    "func_name": "run_measurement"
  },
  "debug": false,
  "print_ramp_data": false,
  "start_func": null,
  "tramp": [
    [
      25,
      10
    ],
    (...)
    [
      25,
      10
    ]
  ],
  "tramp_name": "25_375_dip"
}
```

Listing 3: Default configuration of the eNose Platform platform.

```
{
  "btn1_func": {
    "func_name": "flush",
    "kwargs": {}
  },
  "btn2_func": {
    "func_name": "run_measurement",
    "kwargs": {
      "sample_pump_id": 0
    }
  },
  "debug": true,
  "print_ramp_data": true,
  "start_func": {
    "func_name": "run_measurement",
    "kwargs": {
      "oneshot": false,
      "sample_pump_id": 0,
      "static": false
    }
  }
}
```

Listing 4: Exemplary configuration overriding the default configuration.

The ESP32 is programmed in micropython and the user can operate the device via serial commands over USB or button presses on the front panel. The front panel in Figure 3.21 has two buttons, that can be configured to trigger a measurement sequence that draws a gaseous sample and runs a temperature cycle or perform a reference measurement. The status of the eNose Platform platform is reported with a bicolor (LED). Furthermore, the front panel exposes the USB and micro SD slot

and features up to four 1/16 " Swagelok ports to connect gas sources to be drawn by the micropumps. The temperature modulation cycle can be changed in a configuration file. A standalone device that can be operated handheld follows a program flow pictured in Figure 3.20.

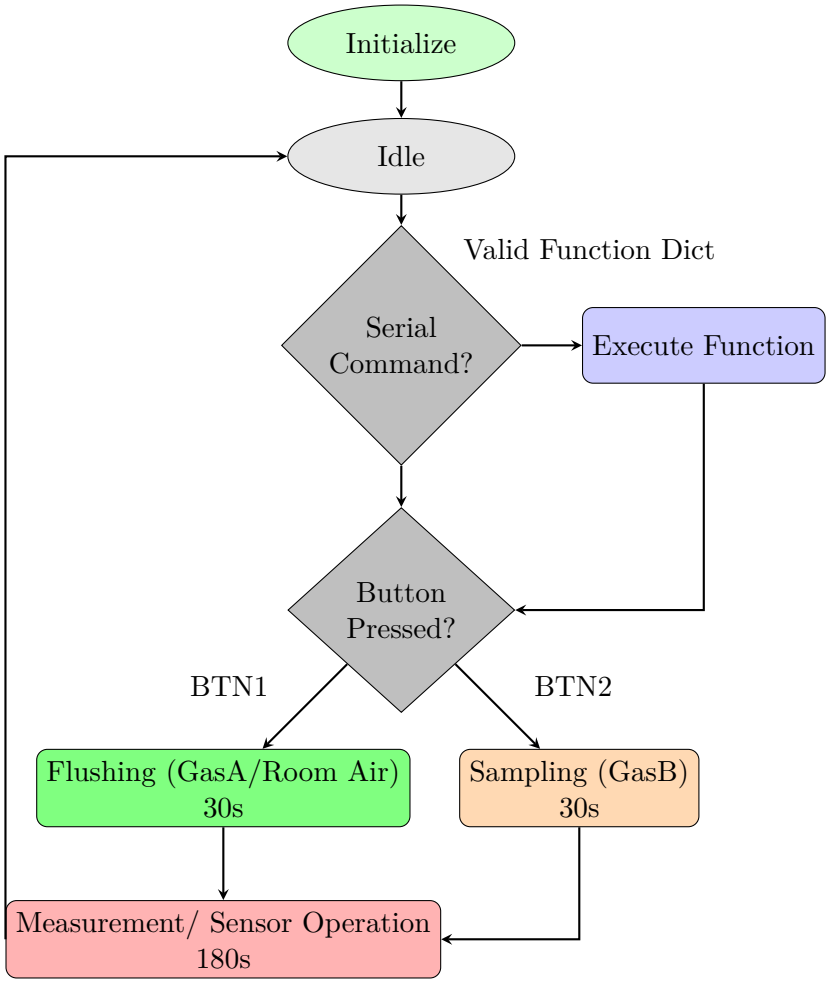


Figure 3.20: Flow chart of eNose Platform platform operation. Green, yellow/orange and red states are displayed by the LED on front panel

After powering/starting the unit, all periphery like pumps, sensor(s) and SD card are initialized. Optionally, if configured so via `start_func`, a certain function is executed directly at start-up. Afterwards, the device enters an idle mode waiting for input. This input could be transferred and scheduled over the serial interface following the same syntax as in the configuration described below. A second control option is given by the buttons on the front panel which are evaluated after the serial commands. The default configuration is supplied in Listing 3. The temperature cycle and its definitions match the default in the esp32ota project (section 3.2.1. The functions executed by buttons `1` and `2` or at start-up are defined in dictionaries with at least the key `"func_name"`. The value like `"flush"` must match a function defined globally. args and kwargs may be included when needed. The bool flag `debug` controls the verbosity of the serial logs/print statements. The data of a whole modulation cycle can be printed with `print_ramp_data` in addition to saving on SD card, e. g. for live results. A list of already implemented functions with their kwargs is given in Table 3.2. More low level functions are documented in the Appendix (see page VIII, Table 6.1).

Table 3.2: Functions implemented on the eNose Platform.

flush(dur=30, temp=None, pump_id=0)
Flush the system with reference gas.

dur:	*int* The time in seconds for which to flush.
temp:	*None* or *numeric* If not None, the temperature of the sensor heater in degrees Celsius, otherwise the sensor resided at room temperature.
pump_id:	*int* Id of the pump to use for flushing, i. e. `0` or `1`. Default `0` (Port A).

measure_sgp(comment="", ramp_id=None)
Measure one cycle of the SGP40(s).

comment:	*str* A comment to add. E.g. a label describing the current sample.
ramp_id:	float If not None, the utc epoch to use as a timestamp, else `time.time()` is evaluated and used

run_measurement(oneshot=True, comment="", static=True, sample_dur=30, flush_after=True, flush_temp=None, flush_pump_id=0, sample_pump_id=1)
Measure one or multiple cycles of the SGP40 optionally sampling gas previously.

oneshot:	*bool* A flag to control, whether a single sample and measurement should be taken. If 'False', these actions will repeat indefinitely.
comment:	*str* A comment to add. E.g. a label describing the current sample.
static:	*bool* A flag to control, whether the sampling period should end before a measurement of the SGP40 is started.
sample_dur:	*int* The time in seconds for which to operate the sampling micropump.
flush_after:	*bool* A flag to control, whether the chamber should be flushed after a single measurement.
flush_temp:	*None* or *numeric* If not None, the temperature of the sensor during flushing, otherwise the sensor resided at room temperature.
flush_pump_id:	*int* Id of the pump to use for flushing, i. e. `0` or `1`. Default `0` (Port A).
sample_pump_id:	*int* Id of the pump to use for sampling, i. e. `0` or `1`. Default `1` (Port B).

To facilitate the flashing and configuration of multiple or different eNose Platform devices, a PyQt GUI has been developed. Screenshots during operation are supplied in Figure 3.22. Rectangle 3.22a shows a number of buttons on the left side to restore or save the current configurations on the computer or attached device. The general configuration itself is situated in the middle of the main window with elements matching those in the json configuration in Listing 3. The firmware and the time of the RTC can be updated via buttons on the right, where the attached device is chosen from a drop-down menu. To aid in the definition of function calls and to document all supported functions, the helper dialog in rectangle 3.22b can be opened with the „Generate Function Call“ buttons. The drop-down menu lets the user select from all implemented functions with text inputs for the kwargs. The docstring of the function is displayed below, together with the finally created json string.

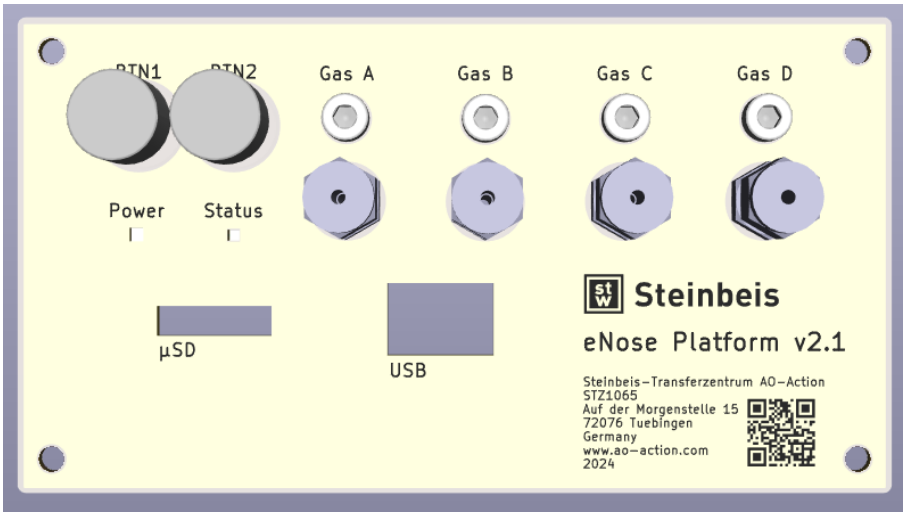
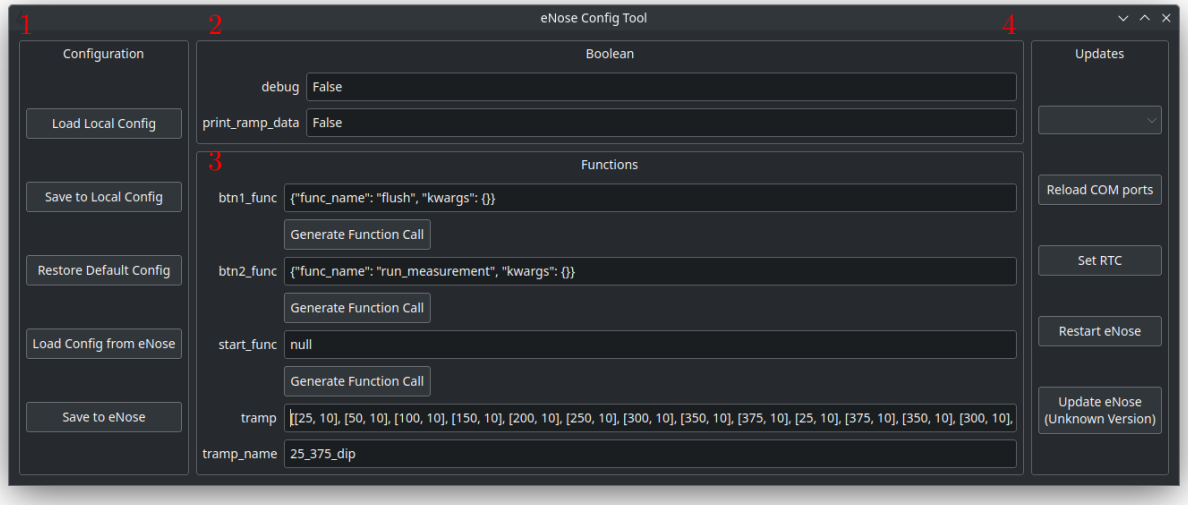
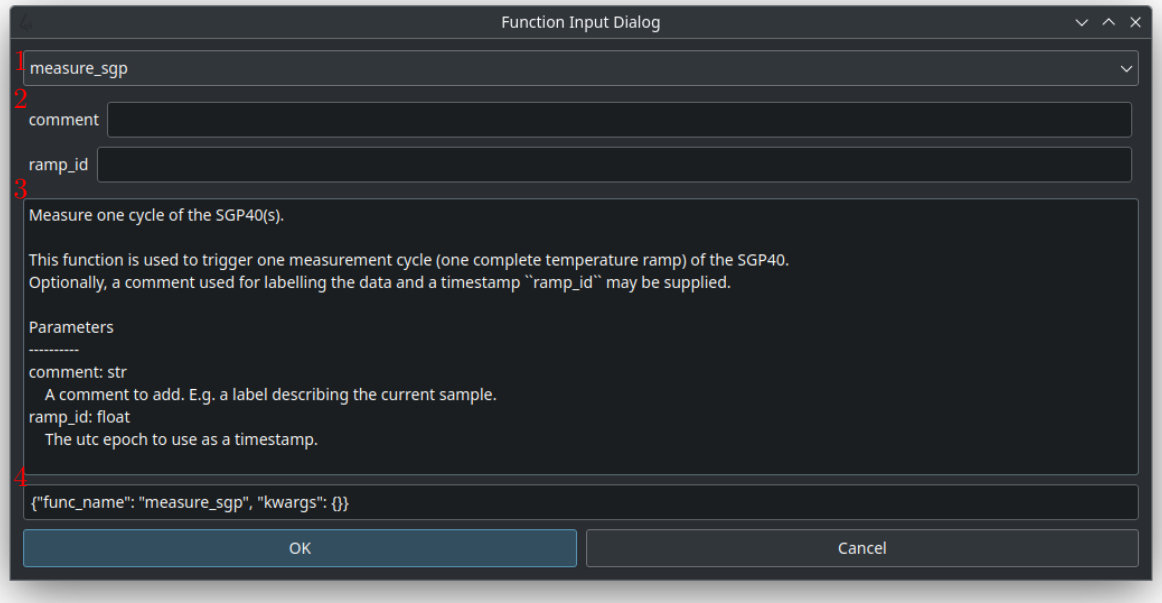


Figure 3.21: eNose Platform platform front panel.



(a) Main window of the eNose Platform configuration GUI.
(1) Config buttons
(2) Boolean flags
(3) Function assignment
(4) Update buttons



(b) Dialog to create a function definition in the eNose Platform configuration GUI.
(1) Drop-down menu to pick an implemented function
(2) Inputs for kwargs
(3) Doc string
(4) Resulting json string

Figure 3.22: eNose Platform Config Tool.

3.3 Data Processing

While most regressors or chemometric models work with feature vectors as input, CNNs can deal with higher dimensional, image-like data. Grey scale images correspond to matrices with rank two, and colored images are tensors with rank three. Therefore, this is the dimensionality of input most commonly fed into CNN frameworks. A gas sensor based on a single SMOX material that is operated under constant conditions would yield a scalar at any point in time, typically a (logarithmized) resistance. An array of n such sensors already produces a vector or length n. Considering a time series of measurement points as a single observable adds another dimension, most prominently stemming from a temperature modulated unit. Thus, the typical temperature cycled SGP30 or SGP40 units with four pixels in this work produce a matrix with the shape (180×4). For the input of a CNN this is exploded into the shape $(180 \times 4 \times 1)$, whereas simpler models like the multilinear regressions take a flattened vector with the length $180 \times 4 = 720$. The python code snippet in Listing 5 shows the creation of a time series DataFrame made up of such arrays, corresponding to $\mathbf{X}$ and $\mathbf{y}$ data. First, the sensor data `dF_sensor` is merged with the gas/concentration data `dF_gas` on the DatetimeIndex. Next, the DataFrame is verified to only include data from exactly one temperature modulation mode, `ramp_name` of the length `ramp_len`. The associated `ramp_start` index (see section 3.2.1) is used for filtering, discarding any incomplete measurement cycles or those that correspond to more than one unique `gas_condition`. Typically, the set/desired concentrations are taken as the ground truth for the following analysis. Finally, only the last `x` complete cycles per condition are considered to ensure an equilibrated gas atmosphere. Additionally, erroneous conditions with large deviations of set and actual concentrations – e. g. due to en emptied gas bottle – are excluded before this merge operation.

```python
def get_last_x(df_sensor, df_gas, x=1, flatten=True, log_x=True, log_y=True):
    group_keys=['gas_condition', 'ramp_start']

    #make sure only on type of ramp is present
    ramps = list(df_sensor['ramp_name'].unique())
    assert len(ramps) < 2, f"More than one ramp type: {ramps}!"

    # merge with gas concentrations
    merged = pd.merge_asof(df_sensor, df_gas, left_index=True, right_index=True,
                           tolerance=pd.Timedelta(60,'s'))

    # only consider complete ramps per gas condition, a.k.a cleaned
    ramp_lens = list(df_sensor['ramp_len'].unique())
    assert len(ramp_lens) < 2, f"More than one ramp len: {ramp_lens}!"
    ramp_len = ramp_lens[0]

    grouped = merged.groupby(group_keys)
    clean_data = grouped.filter(lambda df: len(df) == ramp_len)
    # group by gas condtion
    grouped = clean_data.groupby('gas_condition')

    # get last x ramps with X and Y data
    lastx = grouped.tail(x*ramp_len)
    midtime_per_ramp = grouped.apply(lambda g:g.index[0]+(g.index[-1]-g.index[0])/2)
    gases = grouped.apply(lambda g: g[gases_keys][0])
    xs = np.stack(grouped.apply(lambda g:(g[sensor_keys].values).values)

    # logarithmize X and Y
    if log_x:
        xs = np.log(xs)
    if log_y:
      gases = np.log1p(gases)
    if flatten:
        data_per_ramp = xs.reshape(xs.shape[0], np.prod(xs.shape[1:])).
    df_merge = pd.DataFrame({'X': xs})
    df_merge.index = midtime_per_ramp.values
    df_merge[gases_keys] = gases.values

    return df_merge
```

Listing 5: Code snippet of the get_last_x function used in the extraction of steady state data points.

Data Sets

For all hyperparameter tuning in this work, independent training, validation and test sets are used (see section 2.3). If not stated otherwise, a third of the whole data set is reserved for unbiased testing. The remaining two thirds are either taken completely for training when no hyper parameter tuning is needed or is split again in the ratio `80:20` into train and validation set, respectively. A custom function for this repetitive task relying heavily on Scikit-Learn's `model_selection` module [73] is supplied in Listing 7. Besides the DataFrame argument `df_merge`, that was obtained by merging $\mathbf{X}$ and `y` data and is to be split, a number of keyword arguments allow for tweaking of the split. `test_size` and `val_size` control the sets' sizes with the just described ratios. The bool flag `shuffle` controls whether the data points in the whole set are shuffled and randomly ordered before performing the splits. This happens by default and is useful for time series data with slow changes or without control over the experiment. For instance, it is beneficial for training to predict the outdoor air quality data, because a wider dynamic range of input (and output) data is presented to the models. Otherwise, a model might learn to predict data in summer, but fails to explain winter's observations. In contrast, the difference between the performance after a shuffled and unshuffled split with homogeneous laboratory data – with similar distribution of stimuli in the first and last of the set/time – can be leveraged to asses the aging behavior of the operated sensors. A model that performs reasonably well after training on a shuffled split, but significantly worse in the unshuffled case is an indicator for an aging process throughout the experiment; given there are not other external changes. Specifying a `random_seed` for pseudo-random splits is useful to obtain the very same split sets – especially for debugging or revisiting of data – and ensures computational reproducibility. The `group_gc` enables the data to be divided in such a way that all points/observables in a single equivalent condition – here called a gas condition or `gas_condition` – end up in the same set. This behavior is ensured by actually splitting conditions and then sorting the data points into the sets accordingly. The size ratio typically stays the same, as the number of measurement points in a condition is quite constant or is constructed that way by extracting `n` steady state points. While not directly straight forward in implementation, this treatment assures that no equivalent data points (with comparably little noise) are both in the train and test (or validation) set, which would be a form of leakage [111]. Tightly linked to the (random) train-test split is the matter of Design Of Experiment (DOE). As most machine learning models can interpolate decently but grossly fail explaining targets outside the range of the input training data, it is desirable to obtain test data with samples that fill the (hyper) space of all independent variables quite well. Here, random continuous concentrations of gases have been mixed when possible, in a random sequence to counteract drifts or other temporal effects otherwise falsely associated with certain conditions. The three main split procedures are visualized in Figure 3.23.
Roughly 25 thousand data points (objects) from the Umweltbundesamt (UBA) data set are split into train, validation, and test sets, represented by orange, green, and red bars respectively. Listing 7 describes the underlying function used to create the splits. The top row shows an unshuffled (`shuffle = False`) split, with all data points of one set in a single sequence. The middle row shows the resulting splits after previously shuffling all data points. A compromise of these two extremes is depicted in the bottom row (`group_gc = True`). Here, the unique experimental conditions – in case of the UBA data whole days – are first divided and split into the three sets. Next, the single data points are sorted accordingly. Consequently, the final sets are not as fragmented.

```python
# controlled laboratory conditions; e.g. GMS data set
df_gms['gas_condition'] = (df_gms[gases_keys].diff().abs() > 0).any().cumsum()

# time series happenstance data, group day-wise; e.g. UBA data set
# add day of DatetimeIndex as Series
df_uba['gas_condition'] = df_uba.index.day
df_uba['gas_condition'] = (df_uba['gas_condition'].diff().abs() > 0).cumsum()
```

Listing 6: Generating a numeric gas_condition based on concentration changes.

The numeric, monotonously increasing `gas_condition` can be introduced as exemplarily shown in Listing 6 for controlled laboratory data or uncontrolled happenstance data. The work flow greatly benefits from pandas' chained methods. Essentially, a Series of boolean values is created, reflecting if a change in any gas concentration occurs – as defined by the list `gases_keys` for laboratory data – for any given index or if a new day is starting – derived from the DatetimeIndex in time series data. The cumulative sum of this boolean series yields the monotonously increasing `gas_condition`. Shuffling time series data can lead to leakage in the case of high autocorrelation. For discrete data x, the autocorrelation can be calculated according to equation 3.7, where k is a certain lag [112]. In the case of laboratory conditions held constant for a certain duration, the correlation of resistances measured in this time is obvious. For uncontrollable data, this has to be tested individually.

$$r_k = \frac{\sum_{i=1}^{n-k}(x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3.7}$$

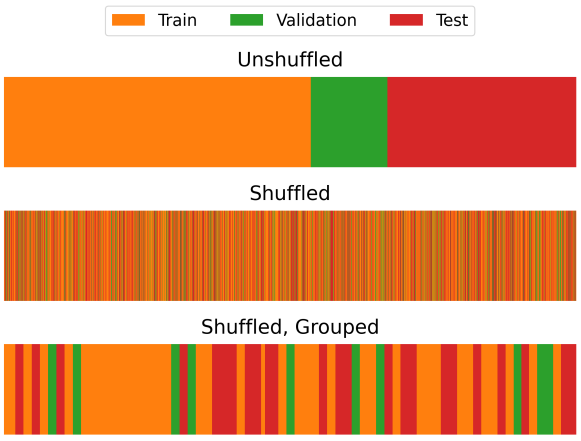


Figure 3.23: Exemplary splits of the UBA data set into train, validation and test sets.

```python
def split_data(df_merge, shuffle=True, group_gc=True, random_seed=0, test_size=0.33, val_size=0.2):
if group_gc:
  # Split into train, val and test set
    gc_per_cycle = df_merge['gas_condition']
    gc_train, gc_test = train_test_split(gc_per_cycle.unique(), test_size=0.33, random_state=random_seed, shuffle=shuffle)

    # And train val split
    # already shuffled, if wanted
    gc_train, gc_val = train_test_split(gc_train, test_size=0.2, random_state=random_seed, shuffle=False)

# create indices
    idx_train = np.where(df_merge['gas_condition'].apply(lambda g: g in gc_train))[0]
    idx_val = np.where(df_merge['gas_condition'].apply(lambda g: g in gc_val))[0]
    idx_test = np.where(df_merge['gas_condition'].apply(lambda g: g in gc_test))[0]

# split based on indeces
    df_train, df_val, df_test = list(map(lambda x: df_merge.iloc[x], [idx_train, idx_val, idx_test]))
else:
     # Split into train, val and test set
    df_train, df_test = train_test_split(df_merge, test_size=0.33, random_state=random_seed, shuffle=shuffle)

    # And train/val split
    # already shuffled, if wanted
    df_train, df_val = train_test_split(df_train, test_size=0.2, random_state=random_seed, shuffle=False)
return df_train, df_val, df_test

```

Listing 7: Code snippet of the split_data function.

4 Results and Discussion

The results presented in this section revolve around two major data sets assessing Outdoor Air Quality (OAQ). The first covers very well defined and controlled gas samples under dynamic laboratory conditions in random sequence and mixtures of continuously random target gas concentrations. The utilized GMS is described in Section 3.1 and this data set is therefore called the GMS data set [99]. The concentrations ranges are listed in Table 4.1 alongside the WHO's recommended short term exposure limits [1]. Figure 4.1 shows the random combinations of any two gases. Each point scattered corresponds to one experimental conditions of 20 minutes. This data has served as a feasibility study to demonstrate whether a single SMOX sensor array can be used to predict the target gases in mixtures by relying on temperature modulation and advanced chemometric models.
In a next step, the second data set features real outdoor air as measured on the campus „Morgenstelle" of the University of Tübingen[1]. The collected data is referenced against the measured O_3 concentration as reported by the UBA as an hourly mean. Hence, this is called the UBA data set [100]. The sensors used in both sets are commercial SGP30 sensors operated in temperature modulation with the esp32ota framework developed for this application (see Section 3.2.1). In the UBA data set, the SGP30's successor SGP40 has been been available and was investigated to screen its suitability to replace the SGP30. Most of the model tuning and comparison has been conducted on these data sets. Different preprocessing schemes have been investigated, and comparison of shuffled and unshuffled timeseries data can be used to address sensor aging.
In a follow-up long term study, the first results about the relevance of different input features created by the distinct temperature cycle have been analyzed with a ridge regression, resulting in the definition of a new modulation scheme. Both major data sets have been used to compare different ML models and further interpret and understand the importance of different temperature regimes in the modulation cycles [113]. The published results are reflected and complemented in Section 4.8.
The next evolution of the UBA experiment approach included the design of a novel modular and self contained sampling unit, described in detail in Section 3.2.3. It has been designed around the SGP40 that has been proven to work well for the prediction of ozone in the UBA data set. These units can be operated offline and therefore be placed in different places and closer to public reference stations.
To aid in bridging the gap between material development and application, two analog sensors were put into action with the developed RPi sensor shield to be operated in a similar fashion as the digital counterparts. They have been evaluated in a limited time frame between October and December, 2024. A home made sensing material – Samarium Iron Oxide (SFO) prepared by Benjamin Junker [114] – on a ceramic substrate with a platinum backside heater as well as a commercial TGS2600 from Figaro were employed.

[1] N48°32.267′ E9°2.299′

One eNose Platform-platform has been deployed on the campus Morgenstelle since May, 2024 and evaluated until December, 2024. The unit has been configured to draw gaseous samples with a maximum flow rate of about 25 mL min^{-1} [115] for 30 s before each measurement cycle without active flow („static“).
In a parallel experiment in collaboration with Trento's Fondazione Bruno Kessler (FBK), one sensor module has been placed in „Parco Santa Chiara a Trento“[2], a park in the city center or Trento, Italy. Data has been available starting September, 2024. Similar to the UBA data set, the publicly reported concentration of pollutants by the European Environment Agency (EEA) [116] were used as the ground truth. However, due to the direct proximity of the reference measurement equipment, also more locally confined target gases have been investigated. Besides O_3, NO_2 and SO_2 have been considered. The author would like to thank Andrea Gaiardo in particular for the fruitful exchange and collaboration and Pietro Tosato for realizing the implementation of the eNose Platform and sharing of the acquired data. The approach presented here is matching their previous work, reporting the monitoring of O_3 with a portable sensor solution [117, 118].

4.1 GMS Data Set – Laboratory Conditions

The GMS experiment was designed to cover random mixtures of pollutants in continuous concentration ranges relevant for outdoor air quality [1, 119, 120] described in Table 4.1. The dosed gases have been broadened from the classical pollutants CO, NO_2, O_3, and SO_2 to include a wide span of RH levels, as humidity can vary widely in most applications and is a well known interfering gas for gas sensors based on SMOX materials [44]. CH_4 is the second most important greenhouse gas [121], that has steadily increased globally from 1,700 ppb in the 90s to around 1950 pbb in 2024 [122]. It thus has been introduced as a background (1,700 – 2,000 ppb). H_2 is added in a random range between 450 and 550 ppb equivalent to the atmospheric concentration. While H_2 is harmless gas in it can be used as a proxy for human presence in IAQ applications [123]. Furthermore it can be considered a potent and important interfering gas as most gas sensors based on SMOX materials are very sensitive to it.

Table 4.1: Concentration ranges encountered in the GMS data and WHO recommended limits [1]. H_2 and CH_4 have been added as interferants not analytes.

Gas	Concentration	WHO limit
CO	300 – 7,200 ppb	30,400 ppb (1 h)
NO_2	20 – 300 ppb	106 ppb (1 h)
O_3	30 – 300 ppb	51 ppb (8 h)
SO_2	40 – 300 ppb	190 ppb (10 min)
RH	5 – 80 %	
H_2	450 – 550 ppb	
CH_4	1,700 – 2,000 ppb	

[2] N46°3.816′ E11°7.547′

Figure 4.1 shows the more than 2.5 thousand unique gas conditions created in the GMS data set according to the concentration ranges in Table 4.1. In each bivariate subplot, every point corresponds to one experimental gas composition described by the the two analyte concentrations on x and y axis. The bar plot histogram with percent resolution on the upper and right border highlight the equal distribution for any one analyte.

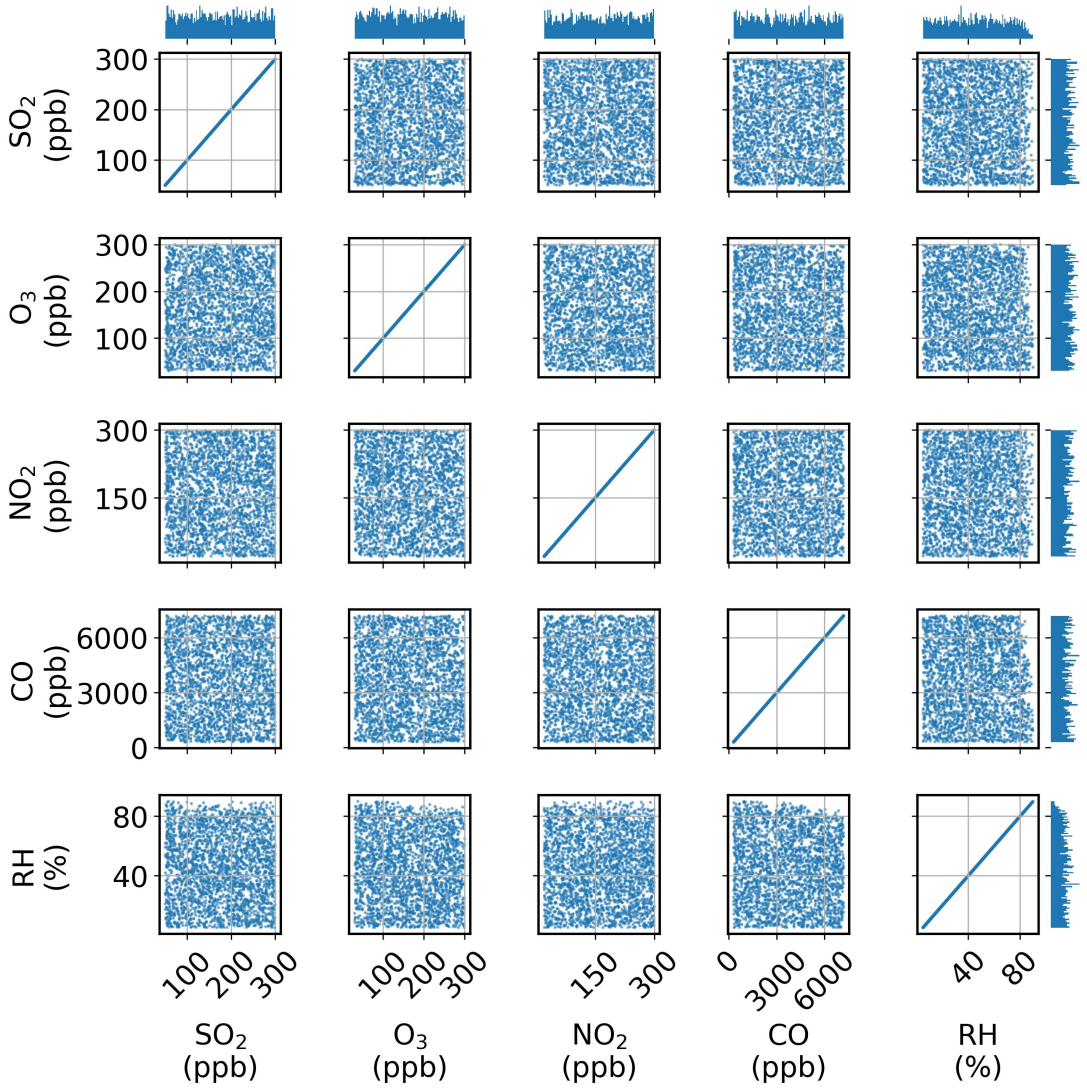


Figure 4.1: Random combinations of target gas concentrations created in the GMS experiment. Every point corresponds to one 20 minute conditions.

4.2 UBA Data Set – Atmospheric Ozone

In contrast to the controlled laboratory conditions of the GMS data set, the ozone concentrations encountered in the UBA data set are described by a happenstance time series. As a secondary pollutant, the ground level or tropospheric concentration of ozone is influenced by a number of factors such as irradiation and other pollutants, but also the stratosphere [4, 124, 125]. The employed strategy to use public reference stations close for field calibration follows approaches by Mueller et al. [126] and by Spinelle et al. [127].

Similarly, Fonollosa et al. could achieve a Root Mean Square Error (RMSE) around 10 µg m^{-3} ($\approx 5-6$ ppb) in the prediction of ozone in multiple places around Barcelona, Spain [128]. The spatiotemporal forecast of ozone by Oliveira Santos et al. reveal that the history of the time series data is of utmost importance for accurate predictions [33]. This has to be kept in mind when creating shuffled data splits to avoid information leakage. The `gas_condition` considered in this data set is one full day, which can not end up in multiple data sets. This aspect is investigated and described in Section 4.5. Figure 4.2 shows time series of the ozone concentration in the full UBA data set from June until August, 2022. The published hourly averages are plotted blue. The zoom-in inset exemplarily shows the resulting cubic spline interpolation for the sixth day in orange.

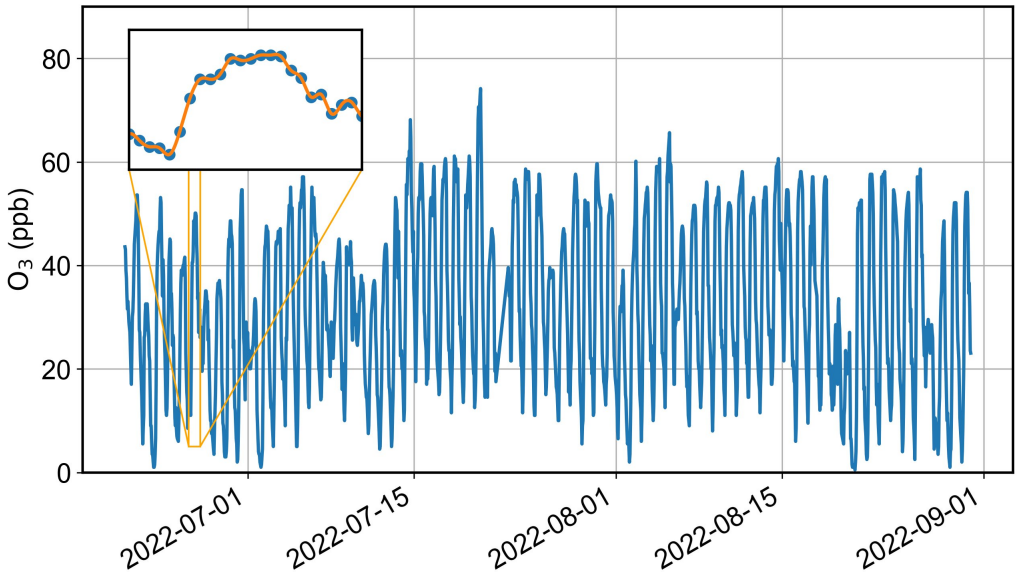


Figure 4.2: Time series of the ozone concentration in the UBA data set.

The violin plots and histograms in Figure 4.3 present the distribution of the ozone concentration in the train, test, and validation set based on a shuffled split that is used for most of the following evaluation. The ozone concentration is described by a two-peaked distribution, sufficiently well represented in the smaller data sets.

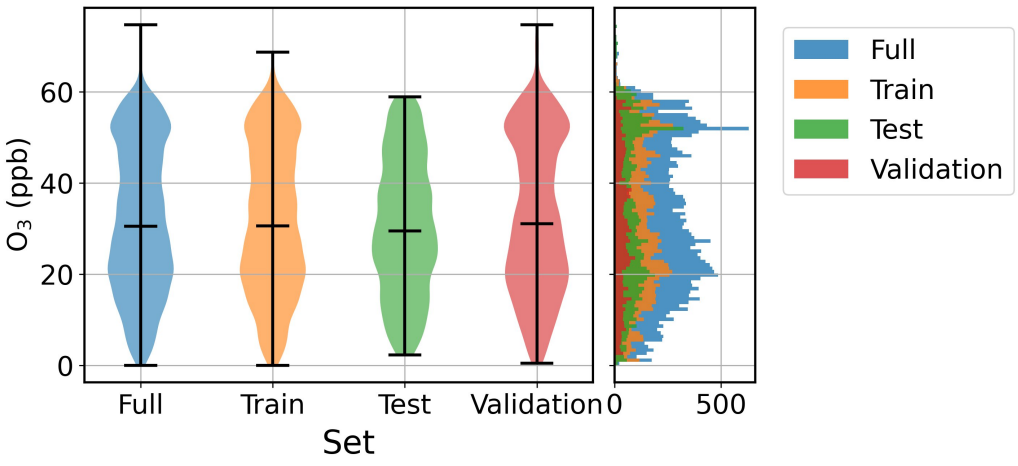


Figure 4.3: Violin plots of the ozone concentration in the data sets of the UBA experiment.

4.3 Model and Data Set Comparison

The findings in this paragraph reflect and complement those reported in [99, 100, 113]. For the two major data sets in this work – GMS and UBA – three types of regression models have been optimized, trained and interpreted: ridge regression, XGBoost, and CNN. The most common type of plot encountered in this work for representing the prediction of trained models is a scatter plot of predicted vs. true concentrations in the test set. Ideal models will yield points following a $y = x$ relation; this relation is emphasized as a red line. Additionally, relative error margins of $\pm 25\%$ are included as dashed lines. The MRE in the test set is added in the legend.

GMS

Figure 4.4 visualizes the performance of the three algorithm types in the GMS data set, individually for each target gas. The general trend is a qualitatively correct prediction for each model and gas. More in detail, the distinct analytes have significantly different accuracies or errors not only due to their different concentration ranges. RH has the lowest MREs for all models around 3 %. This can be understood, as the relative humidity corresponds to rather high absolute concentrations at room temperature ranging from roughly 1,500 to 25,000 ppm. The next highest concentrations are encountered for CO. It is remarkable that low two-digit percent MREs are realized over this large dynamic range. Overall, very few points outside the plotted error regions are observed. The other three gases NO_2, O_3, and SO_2 have been dosed in a similarly low concentration range (up to 300 (ppb)). Out of these, O_3 has the lowest mean prediction error. The SO_2 prediction is most diffuse, that is for one true concentration (range/region) a high variation and deviation in prediction is obtained. For all three gases, it seems likely that the true concentrations fall below a LOD since multiple predictions well above the real concentration occur for small values thereof. The direct comparison of the three architectures reveals the superiority of the CNN model for all of the predictors. The ridge regression often to be found robust comes in second and outperforms the XGBoost regressor.

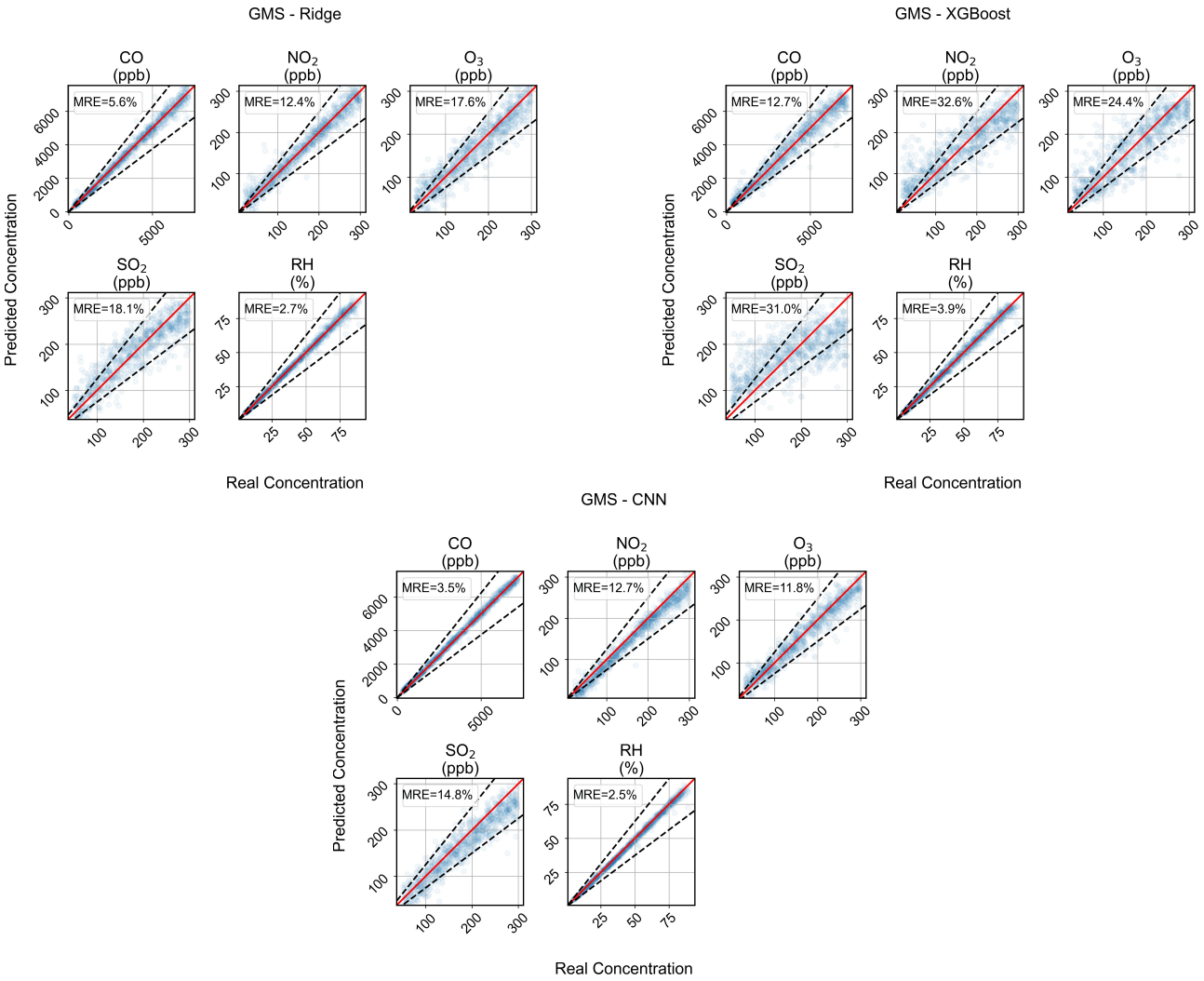


Figure 4.4: True vs. predicted pollutant concentration in the GMS data set for the optimized set of hyperparameters of each model type.

Here, the MRE has been used for the hyperparameter tuning, as it does not have a unit. Other error metrics are summarized in Table 4.2 for the individual gases and averaged over all. Similarly, the Normalized Root Mean Square Error (NRMSE) is unit-less by weighting with the the concentration range ($1/c_{\max} - c_{\min}$). Contrary to the MRE, the RMSE and NRMSE penalize absolutely large deviations more strictly and are not as much governed by errors in the small concentration range that are just large relative to the real value. The reported average Mean Absolute Error (MAE) and RMSE only include the values of the four target gases measured in ppb, excluding RH.

Table 4.2: Different error metrics for the investigated chemometric models trained in the GMS data set.

		Gas	Ridge	XGBoost	CNN
MAE	(ppb)	CO	165.2	329.3	93.0
		NO_2	12.4	30.2	14.0
		O_3	20.8	28.6	14.7
		SO_2	25.6	42.0	21.6
MAE	(%)	RH	1.2	1.6	1.0
$\overline{\text{MAE}}$	(ppb)	excl. RH	56.0	107.5	35.9
RMSE	(ppb)	CO	242.4	441.3	129.4
		NO_2	17.1	39.4	17.6
		O_3	26.9	37.6	19.5
		SO_2	33.4	52.4	27.5
RMSE	(%)	RH	2.1	2.3	1.5
$\overline{\text{RMSE}}$	(ppb)	excl. RH	80.0	142.7	48.5
MRE	(%)	CO	5.6	12.7	3.5
		NO_2	12.4	32.76	12.7
		O_3	17.6	24.4	11.8
		SO_2	18.1	31.0	14.8
		RH	2.7	3.9	2.5
$\overline{\text{MRE}}$	(%)	All	11.3	20.9	9.1
NRMSE	(%)	CO	3.5	6.4	1.9
		NO_2	6.1	14.1	6.3
		O_3	10.0	14.0	7.3
		SO_2	13.4	21.1	11.0
		RH	2.5	2.7	1.8
$\overline{\text{NRMSE}}$	(%)	All	7.1	11.7	5.7

UBA

By and large, the findings about the different models are also applicable to the UBA data set. In particular, all models succeed in explaining the ozone concentrations qualitatively correct. The same order of accuracy exists, with the best CNN model achieving a MRE of 22.5 %. Especially for the CNN and XGBoost it can be seen that lower concentrations are associated with higher relative prediction errors. The MRE and other error metrics are collected in Table 4.3.

The higher MRE compared to that of ozone in the GMS data set is probably owed to multiple factors: First, the reference air quality station has not been in direct vicinity and even included a difference in elevation, often found to be relevant for the distribution of ozone. Still – or even for this reason –, the results presented here are remarkable. Further, the ambient air is obviously not as controlled as laboratory gas atmospheres and is also subject to changes of temperature, humidity, wind, and pressure. Especially in summer time, the lowest residence temperature 25 °C is hard or impossible to achieve as the sensors can only heat and do not have active cooling. Lastly, the higher MRE compared to the GMS data is countered by a smaller RMSE around 8 ppb due to the lower concentration range. This RMSE is in the same order of magnitude as those errors reported by Fonollosa et al. with a similar calibration scheme [128].

Table 4.3: Different error metrics for the investigated chemometric models trained to predict ozone in the UBA data set.

		Ridge	XGBoost	CNN
MAE	(ppb)	6.0	5.9	5.2
RMSE	(ppb)	8.3	8.4	7.8
MRE	(%)	26.3	26.9	22.5
NRMSE	(%)	11.2	11.2	10.6

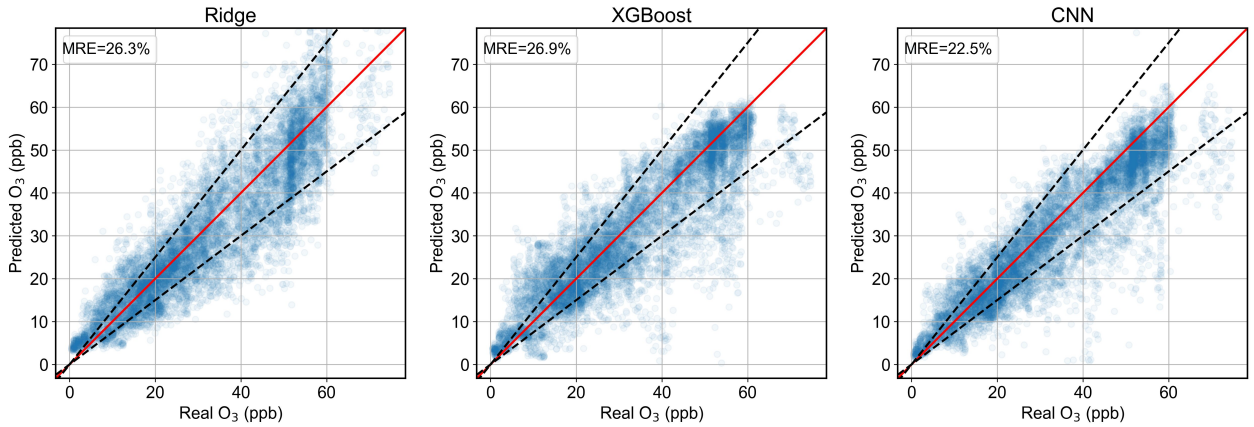


Figure 4.5: Predicted vs. true ozone concentration in the UBA data set for each model type with an optimized set of hyperparameters.

Hyperparameters

As described above and in Section 2.3, a dedicated validation set has been used to assess the best combination of hyperparameters investigated in both the GMS and UBA data set. All possible combinations of parameters have been tested in a grid search, as they might very well be correlated. A smaller number of filters in the convolutional layers might exemplarily require for them to be larger. The possible choices as well as the selection for each data set are collected in Table 4.4. The L_2 regularization has individually been tuned for the kernel (weights), bias, and activity.

Table 4.4: Selection of hyperparameters of the investigated CNN models in the GMS and UBA data set.

Hyperparameter	Values	Selected for GMS	Selected for UBA
Kernel Size	4, 5, 6	6	6
# of Filters in 1st Conv2D Layer	32, 64, 128	32	32
# of Neurons in Dense Layer	32, 64, 128	128	128
L_2 (Kernel, Bias, Activity)	1×10^{-2},		
	$(5 \times 10^{-3}$,	$(5 \times 10^{-3}$,	
5×10^{-3}	5×10^{-3},	1×10^{-2},	
	$5 \times 10^{-3})$	$1 \times 10^{-2})$	
Dropout	0, 1×10^{-2}, 5×10^{-3}	5×10^{-3}	5×10^{-3}
Batch Normalization	True, False	True	True

Alternatively, a univariate evaluation as presented in Figures 4.6 and 4.7 is useful to interpret the choice and importance of these parameters. Here, each panel highlights one hyperparameter. Within this subplot, a single box plot per parameter choice describes the MRE distribution of each target gas prediction for the variation of all other hyperparameters. The common outliers with very high errors, often around 100 %, highlight very inefficient combinations of hyperparameters. In general, the shape and absolute position, e. g. of the median, of the target gases' MREs is qualitatively similar for most parameters in the GMS data set and reflect the fact, that humidity can be predicted with the highest accuracy whereas SO_2 predictions have a higher MRE. The hyperparameter with the clearest beneficial choice is `normalization`: All target gases with the exception of the already accurate humidity prediction have a significantly decreased MRE with batch normalization applied (`True`). For other parameters, no clear indication is observable, either due to a high correlation with others or due to insignificant changes. Still, the medium number of neurons in the final dense layer (128) seems harmful, especially for SO_2, NO_2, and CO. In case of the UBA data set, there are only a hand full of outliers with MREs above 80 % that are not included in the zoomed in plots in Figure 4.7. For all hyperparameters but `normalization` there is little dependence of the MRE observable. Generally, similar trends as in the GMS data set are observed. Interestingly, batch normalization is not selected in the final set of hyperparameters, though on average it lowers the MRE.

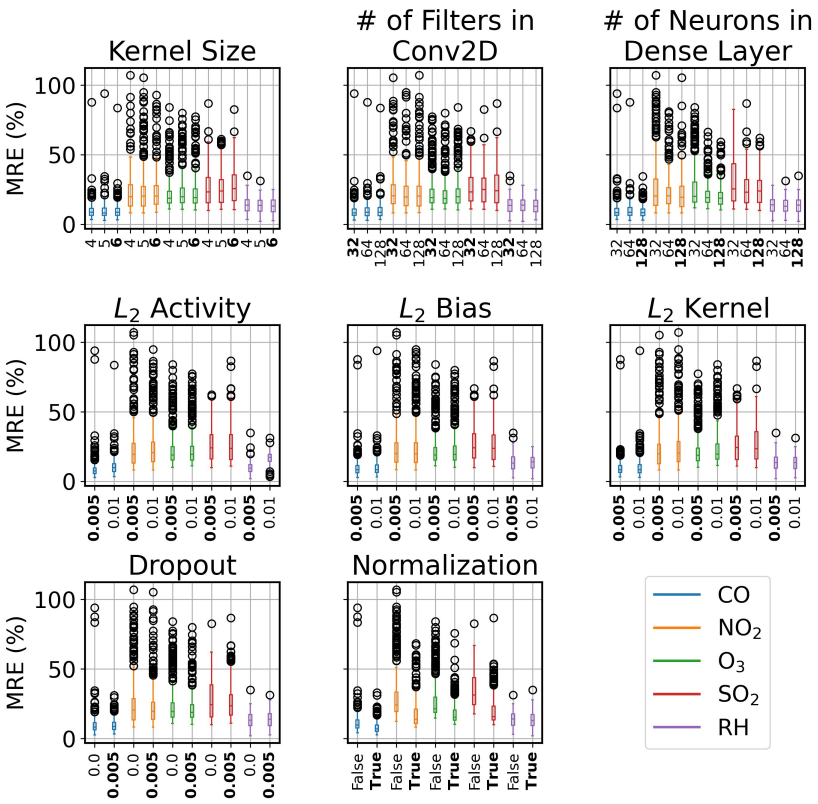


Figure 4.6: Univariate influence of hyperparameters on the MRE of predicted concentrations for the validation set of the GMS data. Selected choices are highlighted in bold font.

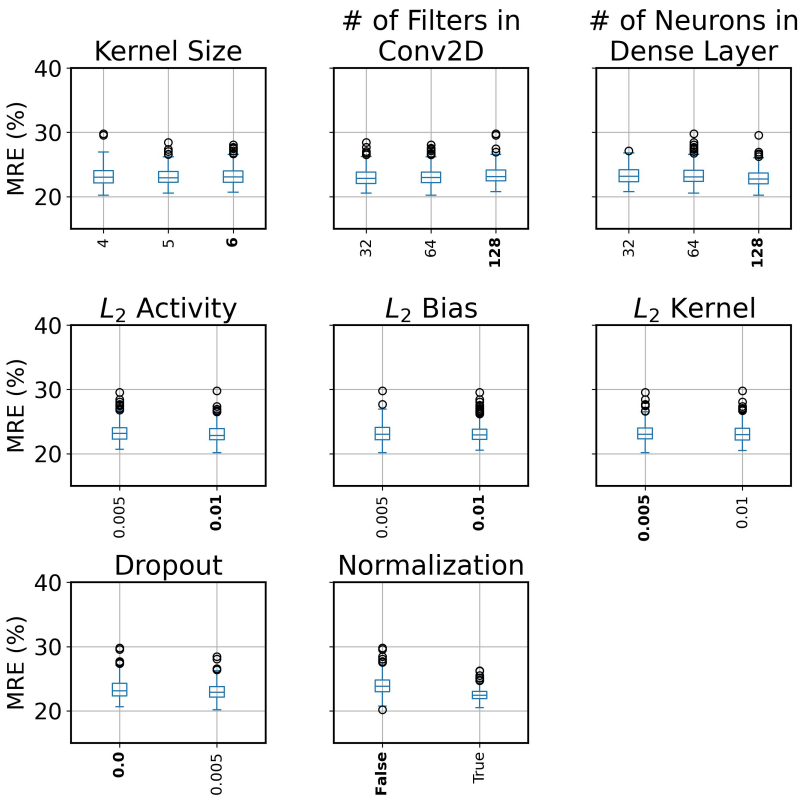


Figure 4.7: Univariate influence of hyperparameters on the MRE of predicted concentrations for the validation set of the UBA data. Selected choices are highlighted in bold font.

The selected hyperparameters for the ridge regression and XGBoost are listed in Table 4.5. The large α value for the ridge regression in the UBA data suggests highly collinear features, as can be understood since the four materials share very basic properties like n-type semiconductivity and the exactly same temperature modulation. Further, the time series resistance does not change instantaneously exhibiting some correlation. The XGBoost regressor requires the hightest number of CARTs to successfully predict the five gases in the GMS experiment; but the lowest number suffices for the prediction of the single target O_3. In contrast, less shallow trees are beneficial for the UBA data. In both cases, reduced subsampling does not seem to help in generalizing to the validation data set.

Table 4.5: Selection of hyperparameters of the investigated ridge regression and XGBoost models in the GMS and UBA data set.

Hyperparameter	Values	Selected for GMS	Selected for UBA
Ridge – α	0, 1×10^{-3}, 1×10^{-2}, 1×10^{-1}, 1, 1×10^{1}	1×10^{-1}	1×10^{1}
XGBoost – Depth	6, 12, 24	6	12
XGBoost – Subsampling	1, 0.8, 0.6	1	1
XGBoost – Number	100, 200, 400	400	100

Weight Comparison - Ridge Regression

The simpler linear models like the ridge regression have few trainable parameters – most notable the slope or coefficient per input feature – that can be compared across the two data sets. Figure 4.8 shows the absolute value of the 720 coefficients, color coded by pixel, of the best performing ridge regressor in the UBA data and those of ozone in the GMS data. To account for the vastly different range of resistances due to different SMOX materials and temperatures, the coefficients have been scaled by the inverse of the standard deviation of the respective logarithmized resistances, the features. The most dominant coefficients are produced in both data sets for pixel 4. However, for the GMS data set, the contribution is more balanced across all pixels and the weighted coefficients are generally lower by almost one order of magnitude compared to the UBA experiment. Besides a few large spikes at low temperatures, especially at the beginning of a cycle, the coefficients concentrate around the „dip“ in the middle. The trend is qualitatively matched by the UBA model with local maxima at the edges of a temperature step, that is around a temperature change. The transient response appears to be decisive. In contrast, the UBA model has very small scaled coefficients at low temperatures at the start and end of the cycle. As hinted at before, these features could be affected by ambient temperature changes more significantly, which have not occurred in well controlled laboratory conditions. Consequently, a version of this cycle truncated to higher start and end temperature `75_375_dip` has been proposed for continued operation and has been used for the long term evaluation in Section 4.7. To keep the overall feature count of 180 measurement points constant, the „dip“ has been broadened.

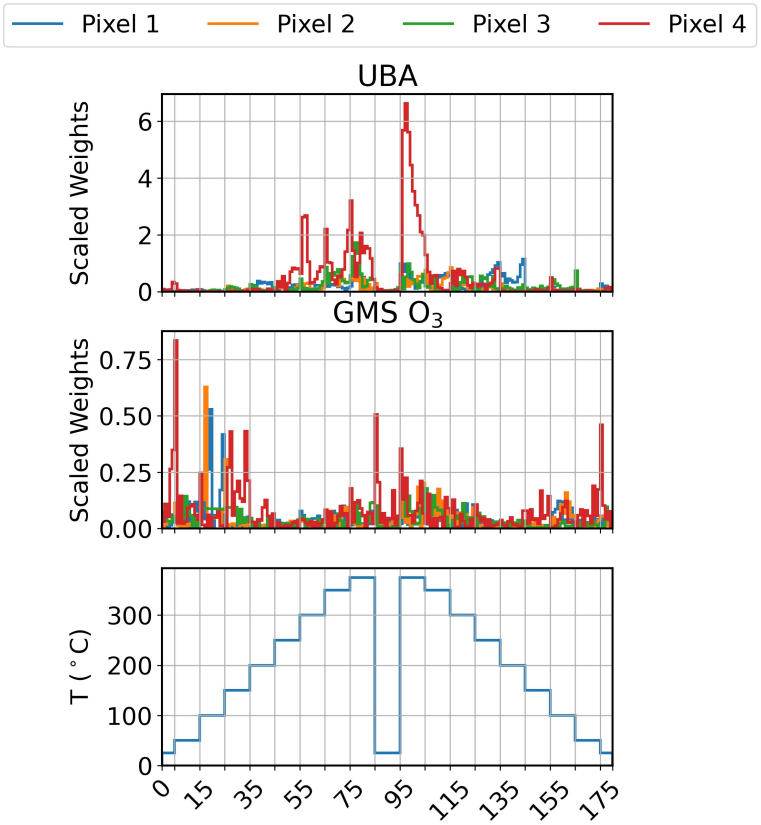


Figure 4.8: Comparison of weights in ridge regressions trained on the GMS and UBA data scaled by the inverse of the standard deviation of the respective feature.

4.4 Limit of Detection

Since both the GMS and UBA data set do not include blank samples, either by design or circumstances, different means have to be turned to for evaluation thereof. Here, the predicted target gas concentrations are defined as a sensor signal of the analytical process as a whole including the chemometric model, most prominently CNN. A linear fit of the predicted values in dependence of the actual concentrations is performed on the test set data. The LOD is derived from the Prediction Interval (PI) expressed in the domain of real concentrations to yield a more conservative estimation than with the Confidence Interval (CI). Figure 4.9 contains the already known true vs. predicted scatter plots with the addition of the PI ($\alpha = 5\,\%$) of the OLS regression. The upper bound for the lowest encountered concentration is translated into an LOD in the original concentration domain, collected in Table 4.6. The prediction of O_3 in real atmospheric air in the lower right panels shows the broadest PI relative to the evaluated concentration range reflecting the higher MRE. Still, the UBA set's LOD for ozone is much smaller than that estimated in the GMS set. However, both sets have very little overlap in terms of concentration. In general the estimated LOD is largely influenced by the overall evaluated range of concentrations. It should be noted that these estimated LODs do not comply directly with DIN32645 [76] but yield a useful metric.

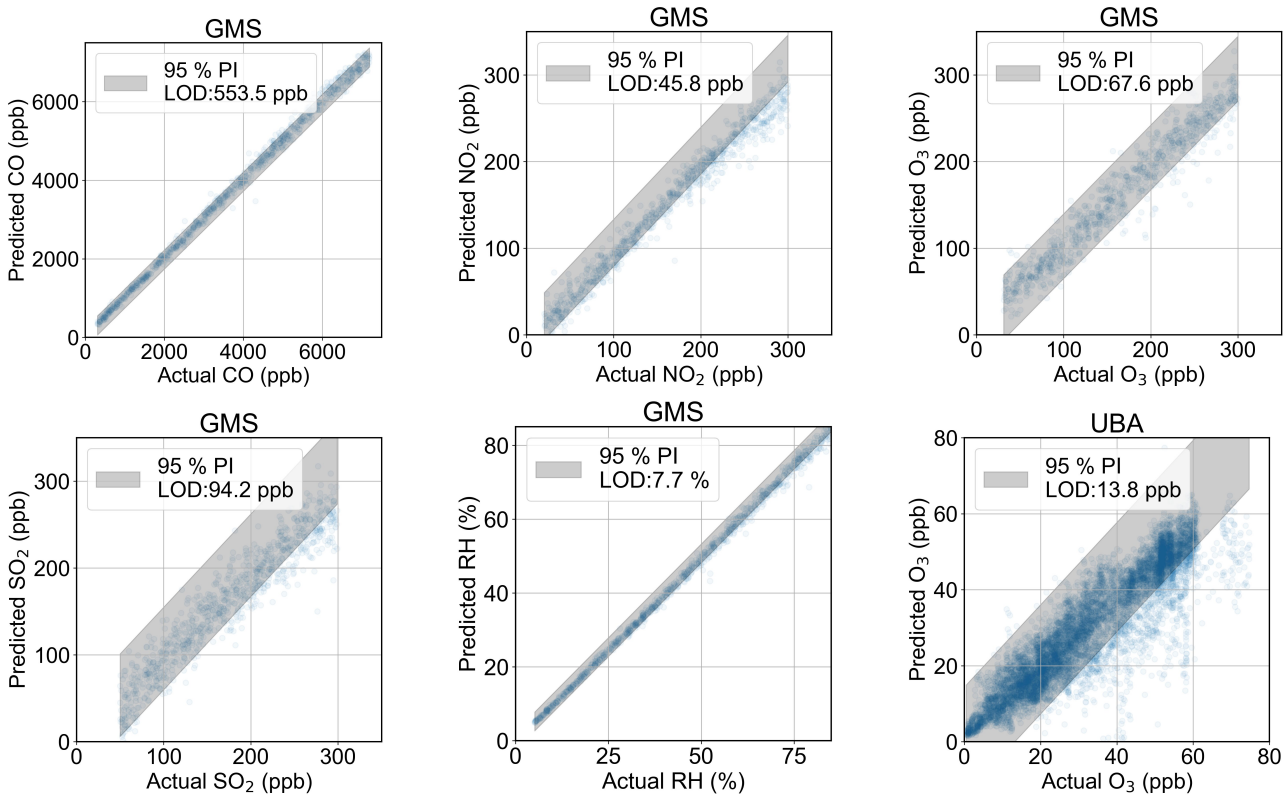


Figure 4.9: Estimated LODs based on predicted vs. true concentration for the respective CNN models and gas.

Table 4.6: LODs of the CNN models for each gas in the GMS data set and ozone on the UBA data set.

Gas	CO	NO_2	O_3	SO_2	RH	O_3 – UBA
LOD	553.5 ppb	4.58 ppb	67.6 ppb	94.2 ppb	7.7 %	13.8 ppb

4.5 Data Treatment

Clean data without flaws or bias is even more important than a sufficiently large body of data in deriving knowledge or fitting models as per the motto „garbage in, garbage out“ [129]. After sight of the data in Figure 4.1 and 4.2, data cleaning and processing is of vital importance. In this paragraph, different pre-processing methods as well as a validation scheme of the overall methodology are discussed.

Happenstance Time Series

When creating or collecting data to train a chemometric model on sensor data, one does not always have access to controlled and reproducible experimental conditions. The UBA data set for instance is a time series of happenstance data.

Importance of Data Set Size

To investigate the importance of the sample size in such coincidental data, the limited UBA data set with a total duration of roughly 2 months has been truncated into different train set sizes in multiples of 10 % (≈6 days) of the whole data set at a stretch. For each fraction, a gap of 5 % (≈3 days) directly following has been added before taking the next 10 % as a test set. That is, the models should learn to predict data in the future with limited correlation between train and test set in a reproducible fashion. The smaller fractions can be fit into the whole data set multiple times and have been moved with a step size of 5 % to smoothen the effect of potential singular extreme events or outliers. The MRE in the test set has been averaged over the different window positions yielding a more robust metric per training size. Figure 4.10 shows all possible splits created in this moving window fashion for the smallest (10 %), largest (80 %), and an exemplary intermediate train set size (30 %). The resulting averaged MREs of ozone prediction in the test sets are plotted vs. the train set size in Figure 4.11. Error bars reflect the standard deviations and the dashed line denotes the MRE for the optimized CNN with a shuffled complete data set. For training set sizes of 30 % and below, the MRE is well above 40 % with an extraordinary large standard deviation. This indicates, that the trained models perform quite well for some of the window positions – i. e. time periods – whereas they fail for others. With such short time frames it is impossible to create a model, that is able to generalize to future events, even in this limited data set spanning over one season only. On the other hand, models that are trained on at least 40 % of the whole data set – corresponding to almost a whole month – approach MREs closer to 30 % on average. The final 22.5 % of the CNN trained on shuffled data are not reached on average but are within the expected uncertainty.

While these results suggest that one month of continuous data is sufficient for training, this is strictly just valid for one sensor unit in one place – with a given pollution dynamic – and this measured season. It is desirable to investigate the applicability in other places, with other target gases, and all seasons. Especially O_3 concentrations are expected to be much lower in winter months, potentially around or below the LOD.

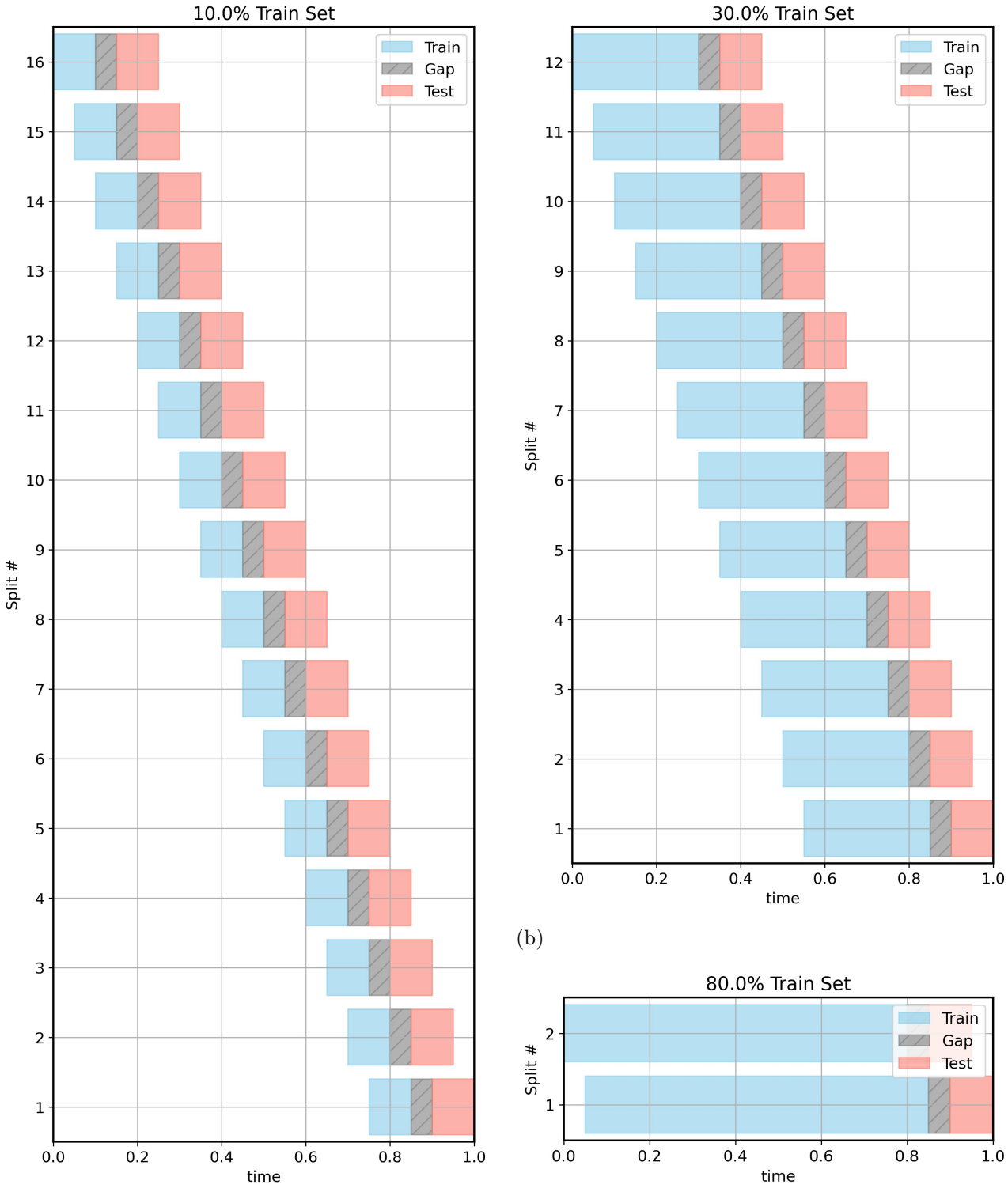


Figure 4.10: Exemplary k-fold splits for the UBA data data set with a fixed temporal distance (5 %) to the train set (10 %) of different test sizes. Depicted the following train set sizes to be evaluated:
(a) the smallest set– 10 %, $k = 16$
(b) a medium sized set– 30 %, $k = 12$
(c) the largest set – 80 %, $k = 2$

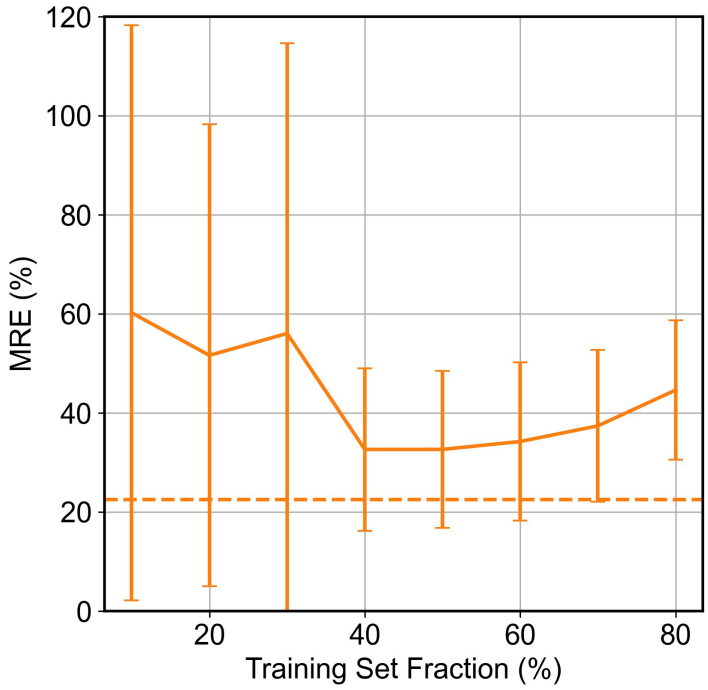


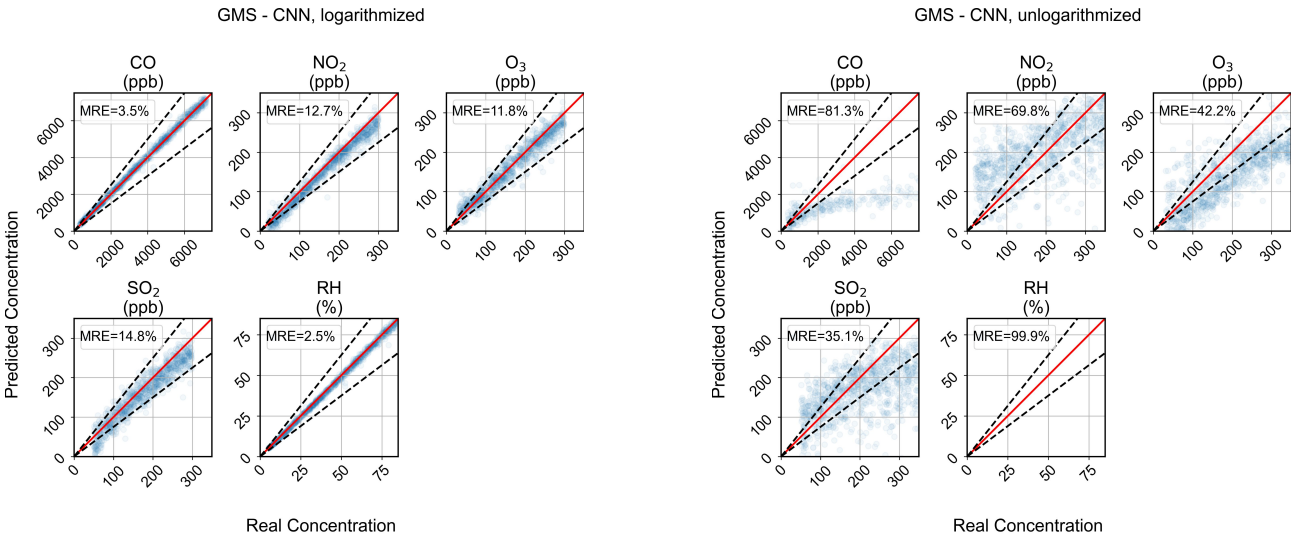
Figure 4.11: MRE in the UBA test data set depending on the train set size.

Linearization

The most common linearization scheme for SMOX sensor readouts or signals is the double logarithmic representation as reasoned and described in sections 2.1 and 3.3. In the following paragraphs the models' (CNNs') performance as evaluated above is compared to the same architecture trained on raw data for the GMS and UBA data set. That is, the optimized set of hyperparameters is kept and only that one model is trained again to predict unlogarithmized concentrations based on raw resistance inputs.

GMS

The superior performance on models trained on (doubly) logarithmized data becomes immediately obvious in Figure 4.12 (a). Not only are the predicted values scattered further apart – with a higher standard deviation – for raw data (b); but the center of the predicted point cloud is also sloped incorrectly, giving rise to overprediction of low concentrations and underestimated high values. The combined effects are reflected by significantly higher MREs. For the high range of absolute concentrations of CO and RH, the predictions fails completely.

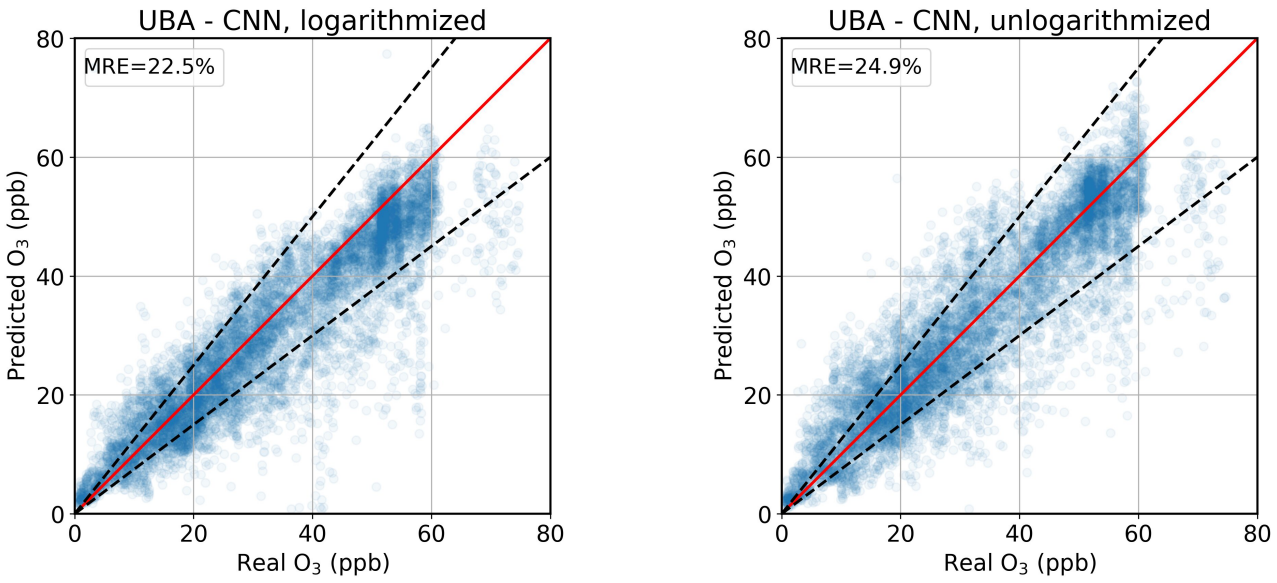


(a) (b)

Figure 4.12: True vs. predicted ozone concentration in the GMS data set with a CNN trained and evaluated on
(a) logarithmized $\mathbf{X}$ and $\mathbf{y}$
(b) raw $\mathbf{X}$ and $\mathbf{y}$

UBA

While the CNN models in the GMS experiment are benefiting from the log-transformation, this preprocessing is absolutely crucial for the training of a CNN to predict atmospheric O_3 in the UBA data set: The hyperparameter tuning excluded batch normalization from the layer sequence. In turn, unlogarithmized, raw data causes exploding gradients making the optimization of such a model impossible. The models compared with logarithmized and raw data inputs in Figure 4.13 therefore differ slightly in their architecture as batch normalization is re-added in the case of raw data. In line with the GMS data, the model trained on raw ozone concentrations in Figure 4.134.13b yields a higher spread of predictions for high concentrations and larger absolute deviations or outliers. Yet, the unlogarithmized data does not cause a dramatically larger MRE. The reason likely lies in the potentially heteroscedastic nature of the predictions and associated errors. Hence, the residuals of prediction (predicted - true concentration) are plotted vs. the predicted concentration in Figure 4.14. The Breusch-Pagan [130] and White [131] tests suggest heteroscedasticity for both cases [132]. Still, the absolutely higher residuals in the raw data case are obvious and the residuals in the logarithmized instance appear more constant with a slight decrease towards higher predictions. These findings are in line with the well established and understood chemometric models, that work best for the explanation of linear relations. Moreover, the log-transformation is often useful for dealing with highly dynamic data, spanning multiple orders of magnitude, as is the case for the resistance of temperature modulated SMOX materials.



(a) (b)

Figure 4.13: True vs. predicted ozone concentration in the UBA data set with a CNN trained and evaluated on
(a) logarithmized **X** and **y**
(b) raw **X** and **y**

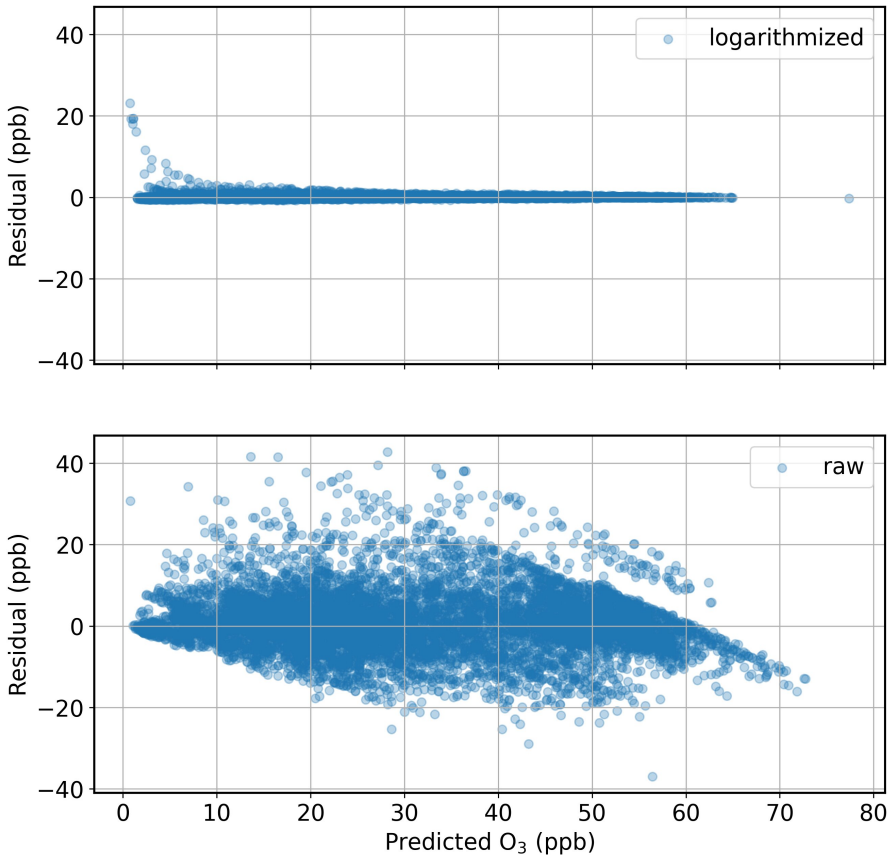


Figure 4.14: Residuals of the predicted ozone concentration in the UBA data set.
Top: logarithmized **X** and **y**
Bottom: raw **X** and **y**

Shuffling

As described in Section 3.3 and hinted at in the investigation of train set size importance in the paragraphs above (see page 65), shuffling the whole data set can have a significant impact on the performance of chemometric models. In happenstance time series data, one should note that shuffling can lead to a certain amount of leakage from the train to the test set, as most time series' are somewhat autocorrelated. That is, most points in time are correlated to the previous time period of a certain duration. Therefore, randomly sampling or selecting small time windows from the whole data set may effectively include most of the variance. The autocorrelation of the UBA data set – for the hourly mean concentrations – is laid out in Figure 4.15 using the `statsmodels` package [132]. Here, the averaged correlation of data points with a certain lag or time difference is plotted as stems together with the 95 % confidence interval in a cone shape. It can be observed that the correlation for lags larger than 3 days is not statistically significant anymore. Thus, this is exactly the gap length introduced between the train and test set in the investigation of train size importance. The implication for a shuffled train-test split of the UBA data is, that some leakage can occur do to each daily `gas_condition` being somewhat correlated to the previous days which could end up in a different set.

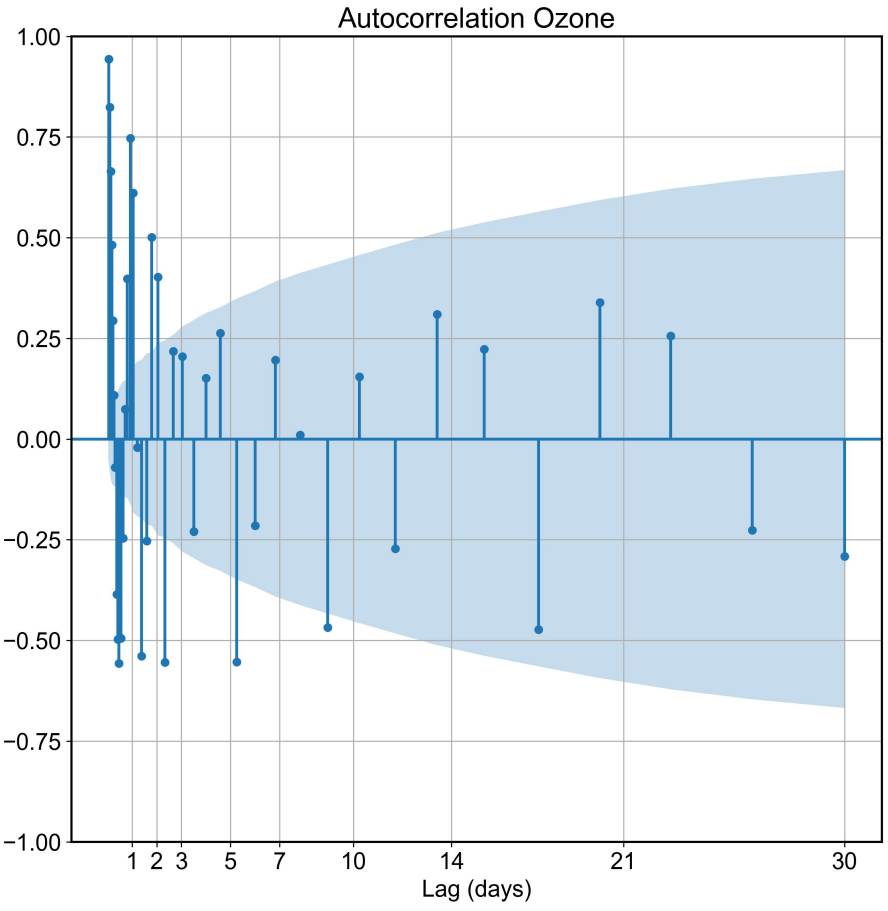
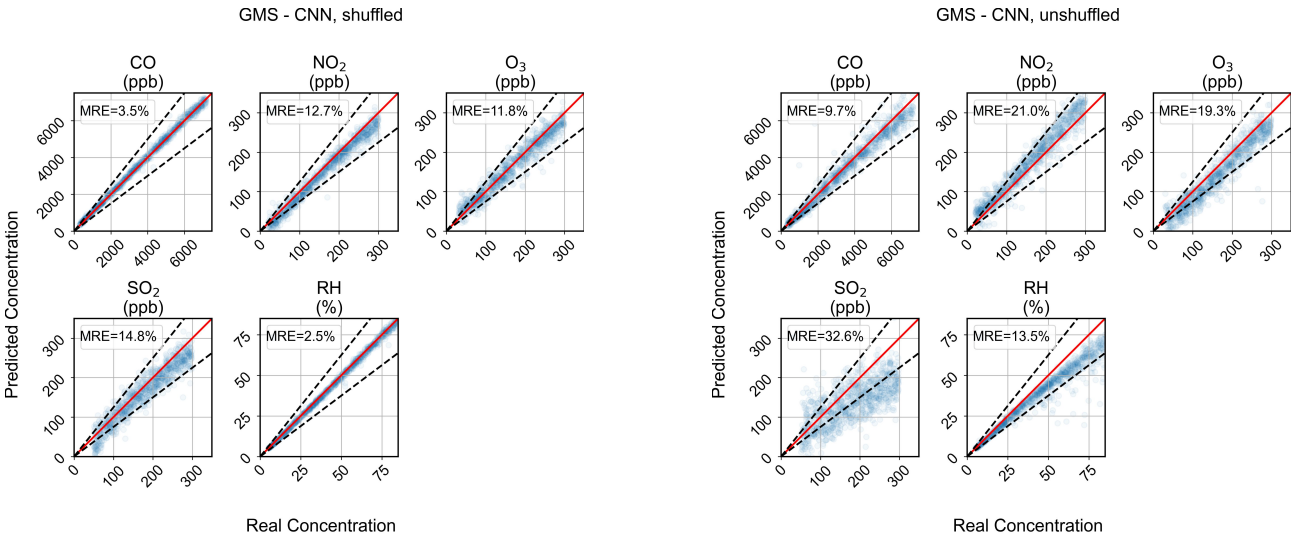


Figure 4.15: Autocorrelogram of the UBA data set. The evaluated lags are logarithmically spaced.

GMS

Figure 4.16 compares the accuracy of predictions for a shuffled (as before) and unshuffled data set split. For CO, O_3, RH the predicted vs. true plots stay qualitatively the same, while the MRE is significantly increased. For NO_2 and SO_2, the two analytes with the highest MRE, the curve changes shape or slope. High concentrations of NO_2 tend to get overestimated, whereas high concentrations of SO_2 are underestimated even more severely. When considering the laboratory conditions to be quite reproducible or stable, these findings suggest an aging of the sensing materials. When later conditions, where alteration has already occurred, is included in the training set – as is the case with shuffling – the model learns to compensate these changes.

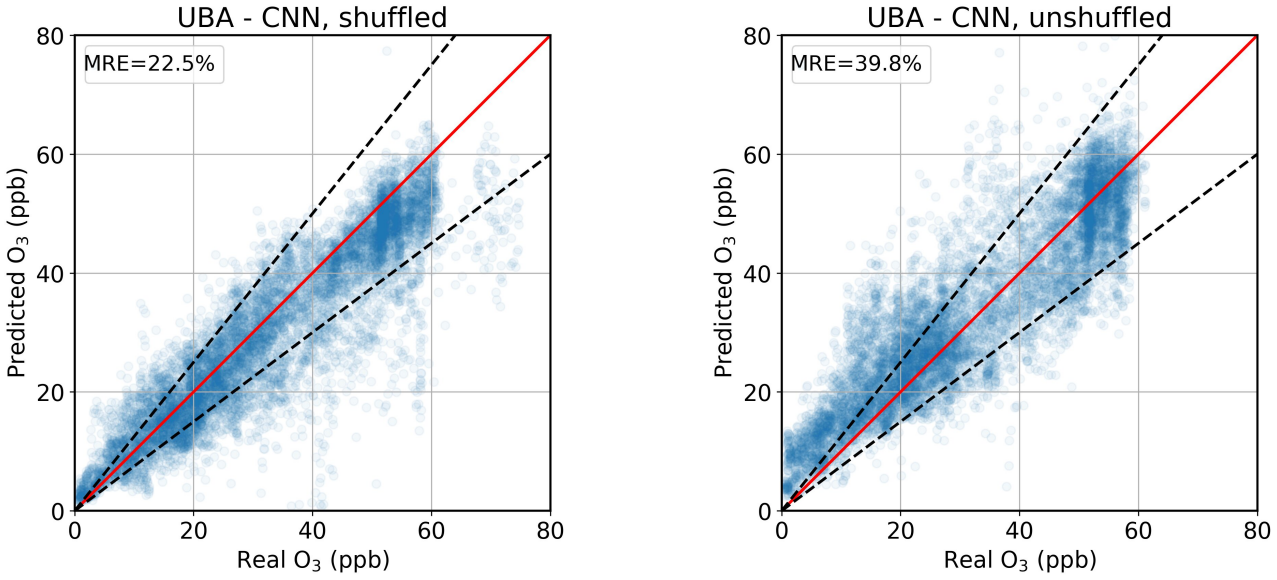


(a) (b)

Figure 4.16: True vs. predicted ozone concentration in the GMS data set with a CNN trained and evaluated on data set split that has been
(a) shuffled
(b) unshuffled.

UBA

The comparison in Figure 4.17 shows similar trends as observed in the laboratory. On top of that, the difference between the shuffled (as before) and unshuffled data set split is even more significant. While aging may definitely occur with these sensors, the ambient atmosphere also changes with the progress of the seasons. A model trained on summer conditions cannot be expected to perform well during winter and vice versa. The investigated data set does not cover enough time to include drastic season changes, see Figure 4.2. Still, the CNN profits from shuffling, observing samples from the complete time frame. This is a reason, that the unshuffled k-fold splits in Figure 4.10 did not converge to the MRE of the full shuffled split.



(a) (b)

Figure 4.17: True vs. predicted ozone concentration in the UBA data set with a CNN trained and evaluated on data set split that has been
(a) shuffled
(b) unshuffled.

Validation of CNN

As introduced in Section 3.3 and the paragraphs above, unique data sets have been used for training the models, validating them with their set of hyperparameters, and lastly to test their performance on an unseen data set. This procedure is especially important for building large, complex models like the CNNs. Here, this training scheme, besides other means like regularization or early stopping, minimizes the risk of overfitting. Figure 4.18 shows the evolution of the loss function MSE in both data data sets for the best CNN respectively during training. The blue curve represents the training set used for the optimization of trainable parameters in some form of gradient descent, see Section 2.3. Starting from randomly initialized weights and trainable parameters, one can see that the MSE drops quickly for the training and unrelated validation set in both experiments. Sudden increases or spikes in the loss function witness the overshooting of local minima, often encountered in iterative optimization procedures, and are directly mirrored or even emphasized in the validation set, especially during the first 100 epochs/steps in the GMS. At a certain point, the models' performance does not increase anymore, around 100 steps or epochs. In the UBA data set, the model's loss function of the training set keeps decreasing while that of the validation set stays rather constant. This is an onset of overfitting, where the model fits noise in the training data that cannot be generalized to unseen data, e. g. the validation or test set. In some fatal cases, not encountered here, this process can be joint by an increase of the loss function in the validation set. Overall, these characteristic attributes of the fitting or learning procedure are well described by the MREs in Table 4.7 including metrics for the ridge regression and XGBoost models as well.

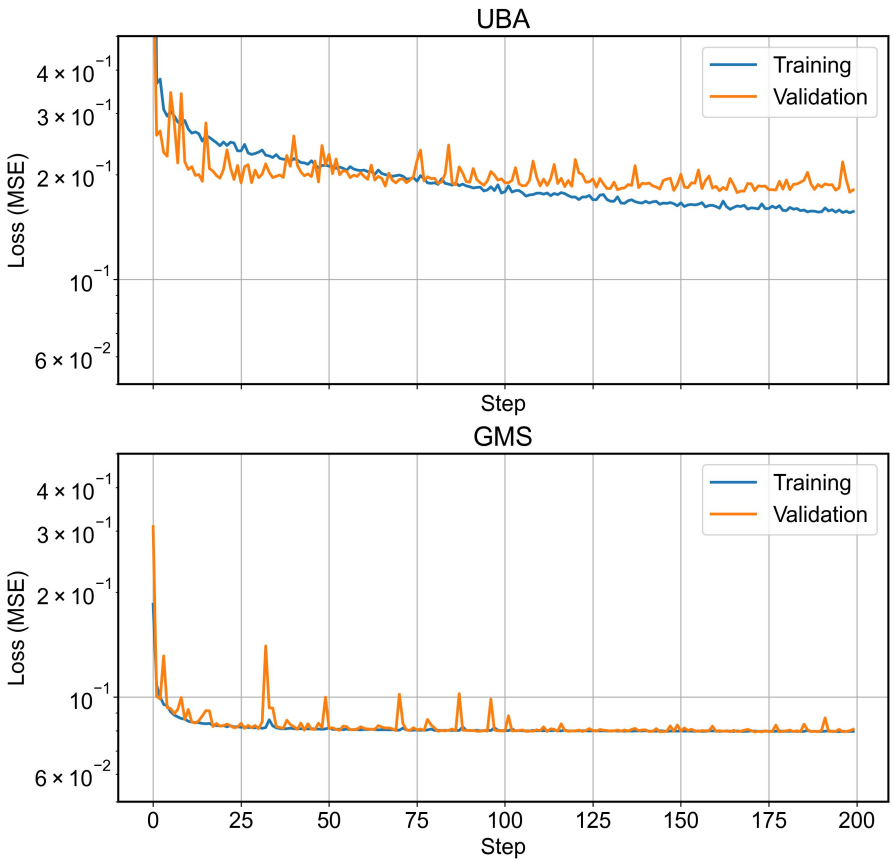


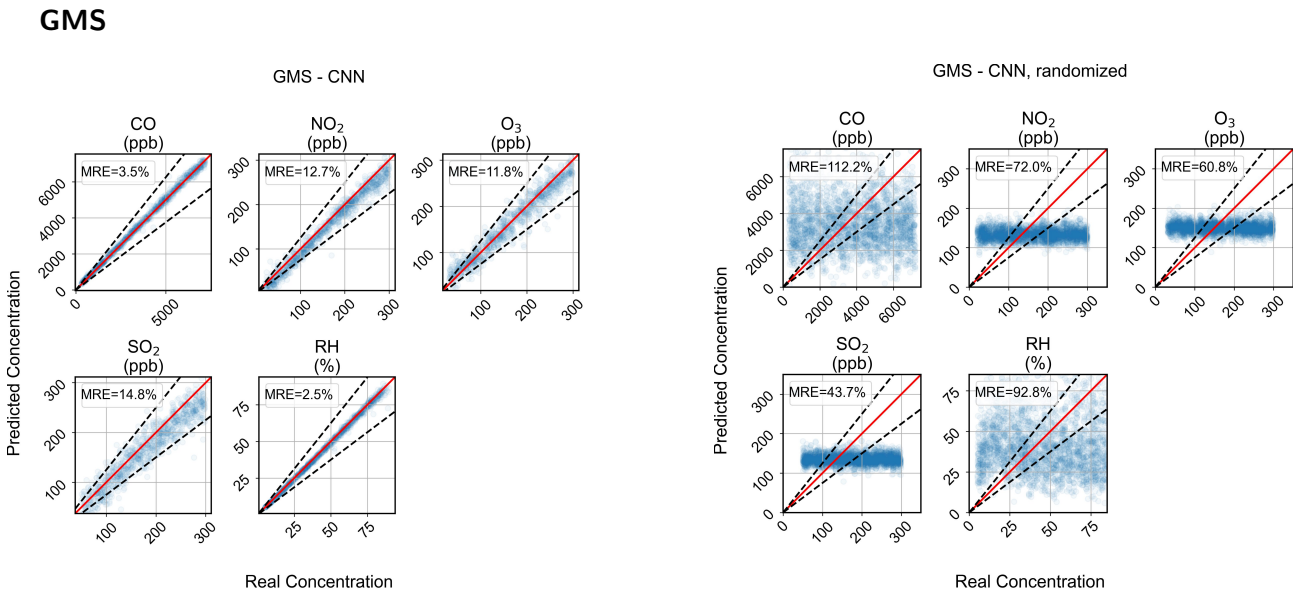
Figure 4.18: Loss MSE in the training and validation set as a function of the steps for the optimized CNN models.

It can be observed that the CNN models produce the most consistent MREs throughout the different sets. The MRE of predicted ozone by the ridge regression in the UBA test set is higher than in both validation and test set. With the XGBoost models, the achieved MREs in the test sets are ridiculously small owing to overfitting. Only the CNNs produce results that are comparable and allow for generalization.

		MRE (%)		
Data Set	Model	Training Set	Validation Set	Test Set
GMS	Ridge	7.0	11.1	11.3
	XGBoost	0.3	21.9	20.9
	CNN	9.0	8.8	9.1
UBA	Ridge	39.7	23.4	26.3
	XGBoost	0.04	25.3	26.9
	CNN	16.1	20.4	22.5

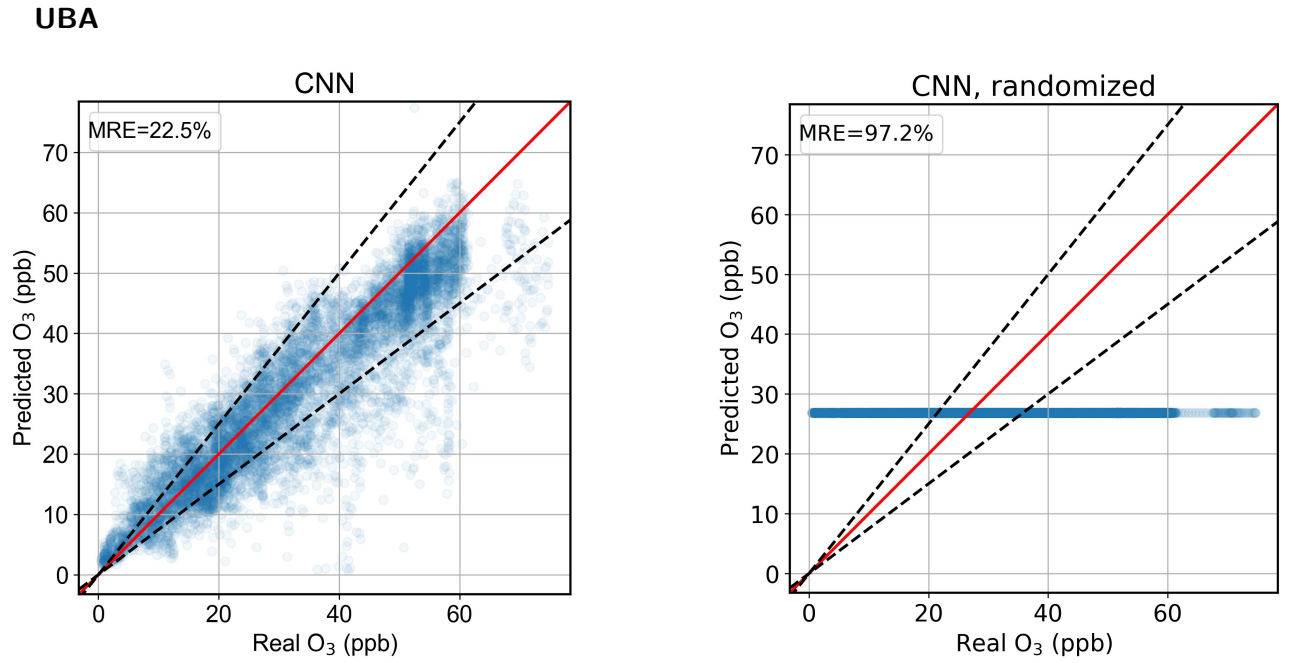
Table 4.7: MREs in the training, validation and test set of the GMS and UBA data with the optimized models.

Besides using unrelated sets for training and testing, a simple method to validate the proper prediction and generalization ability of a chemometric model is breaking the relation between **X** and **y** and re-training the algorithm [133]. The resulting models simply minimize the prediction error by constantly outputting the mean of the data set which represents the expected value and therefore best guess. This sanity check rules out, that more complex architectures – like the CNN used here with its roughly 1.5 million trainable parameters, see Figure 4.27 – simply learn to reproduce the presented patterns or relations as in an overdetermined equation system without fetching the underlying causality. Well performing models in these cases would be an indication for overfitting and/or data leakage. Figures 4.19 and 4.20 show the comparison of the best CNN as before (see Figures 4.4 and 4.5) and one trained to predict randomized relations between measured resistances and actual concentrations. As expected, the models cannot learn a relation between those two properties as they have been arbitrarily assigned. Instead, the mean concentration of the encountered data set is predicted – for instance slightly below 30 ppb in the UBA data. These results rule out the possibility of these complex models learning to predict desired properties without causal relations and highlight once more the need for independent test sets. Furthermore, no significant leakage from train to test set is found.



(a) (b)

Figure 4.19: True vs. predicted target gas concentrations in the GMS data set.
Sanity check comparing the results of a CNN trained on a
(a) regular data set
(b) data set with broken relation between **X** and **y** data, expected to fail



(a) (b)

Figure 4.20: True vs. predicted O_3 concentration in the UBA data set.
Sanity check comparing the results of CNNs trained on a
(a) regular data set
(b) data set with broken relation between $\mathbf{X}$ and $\mathbf{y}$ data, expected to fail

Individual Models per Gas – GMS

This is also the case for a ridge regression. The optimization of a ridge regression yields the same set of coefficients for a certain target gas, independent of other gases to be predicted simultaneously, if the inputs are the same. Numerically it makes no difference whether five target gases are predicted by a single model or individually by five models for a given data set. With CNNs however, the feature extracting convolutional layers can learn to extract distinct features focused on single gases alone. These differently processed inputs could cause a distinct set of coefficients in the regression layers. Consequently, individual models for a single gas have been trained to investigate whether the joint training of all target gases simultaneously produces more robust models with the inherent ability to deal with interfering gases or if separate CNNs prevail by specialized features. Figure 4.21 shows the predicted vs. actual values for the five particular models in the GMS data set. For a comparison with a single, joint model see Figure 4.4 or Figure 4.19a. The five models presented here differ in their hyperparameters that have been tuned as before, but for the respective target gases. The optimized values are collected in the Appendix (see page VIII, Table 6.2).
Overall, the CNN models to predict single gases perform in a very comparable fashion to the holistic one. This is also reflected by the similar choice of hyperparameters. For instance, all models prefer batch normalization. The only gas for which a dropout layer ends up in the final architecture is O_3. It could be argued, that the resulting model for the prediction of SO_2 alone is able to predict higher concentrations better since the obvious underestimation of the common model is not repeated. Interestingly, the SO_2 model also ends up with the fewest neurons in the regression layer.

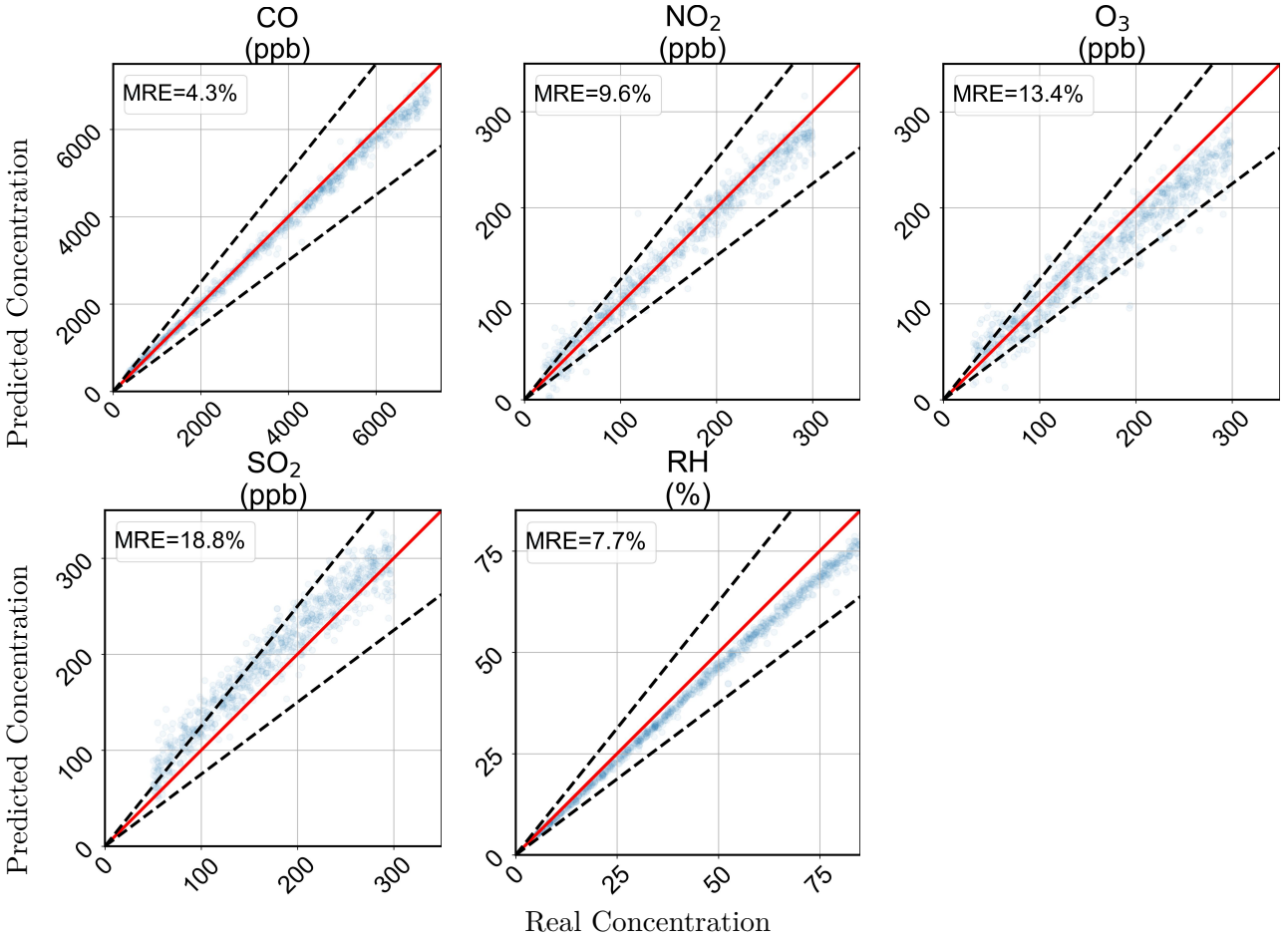


Figure 4.21: Predicted vs. true concentration for individual CNN models trained solely on one target gas at a time, while the others were present in mixtures.

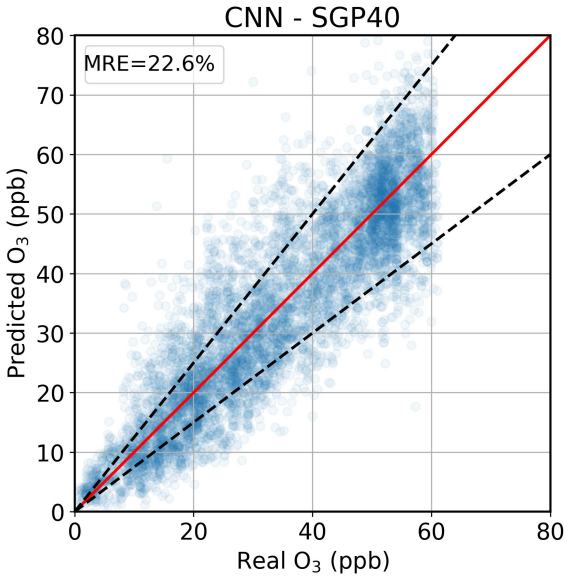
Comparing the MREs in Table 4.8 directly, no large difference between joint and specialized models is observed, but the common model yields better results on average and for all analytes but NO_2. In detail, the CO and O_3 detection deteriorates with single models. The performance of the other two target gases stays rather unchanged. In summary, a joint model for the prediction of all detectable target gases is simpler to implement and handle and needs to be optimized and trained just once. Since no clear deviation, but a slight improvement occurs, it is the preferred scheme for handling gas mixtures, just like for the aforementioned ridge regression without a difference by design.

Table 4.8: MREs of the CNN models trained to predict a single gas in random mixtures in the GMS data in comparison with the model trained to predict all gases at once.

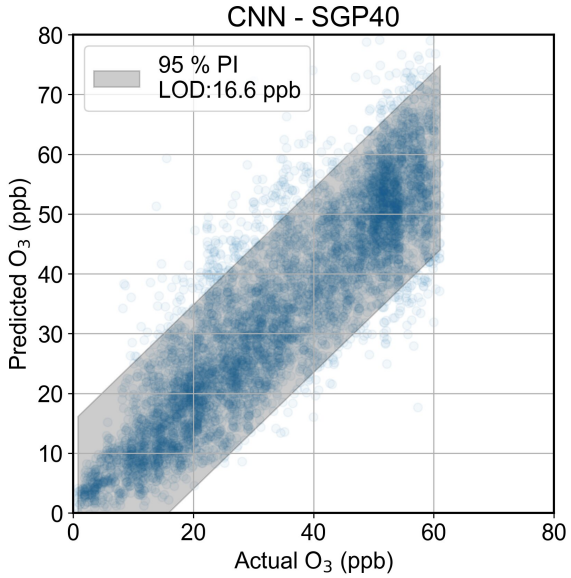
	MRE (%)					
	CO	NO_2	O_3	SO_2	RH	Average
Individual Model	4.3	9.6	13.4	18.8	7.7	10.8
Common Model	3.5	12.7	11.8	14.8	2.5	9.1

4.6 UBA Data Set – SGP40

The previous sections investigated the applicability of temperature modulated SGP30 sensors to predict pollutant gases in laboratory conditions and to quantify real tropospheric ozone concentrations. Further, a number of chemometric models have been optimized and compared in respect to their prediction accuracy. In both scenarios, the CNN models performed best and were able to extract meaningful information from the large feature set. Due to the EOL status of the SGP30, its successor SGP40 should be used for new developments and has been operated in parallel in the UBA experiment. For the same reason, the group's internal reference sensor unit (see Section 3.1) and the eNose Platform platform (see Section 3.2.1) are built around the SGP40.
Figure 4.22 shows the typical predicted vs. true plot (a) and the evaluation of the LOD (b) for the test set, after hyperparameter tuning. The investigation of hyperparameters – analogous to the SGP30 – can be found in the Appendix (see page IX, Figure 6.2 and Table 6.3). A very similar set of hyperparameters has been selected; the smaller number of neurons in the dense layer is made up by a larger number of filters in the first convolutional layer. Both, the MRE and LOD manifest an equivalent performance of the SGP40 compared to the SGP30. The relative error is virtually identical, but the LOD of 16.6 ppb is slightly higher than that of the SGP30.



(a)



(b)

Figure 4.22: Predicted vs. true ozone concentration based on an SGP40 and CNN.
(a) with 25 % error boundaries
(b) with 95 % prediction interval and estimated LOD

4.7 Long Term Stability

The coefficients of the ridge regression – specifically in the UBA data set – were used to define a temperature modulation cycle with a step-wise triangular shape between 75 and 375 °C and an abrupt dip to 25 °C in the middle, expected to yield a feature rich transient response for the prediction of tropospheric ozone. This operation mode is envisioned to be rather robust against ambient temperature changes due to its elevated start and end temperature.

The same SGP40 sensor unit from the UBA experiment has been operated between December, 13th 2022 and September, 11th 2024 in the same manner as in the UBA data set, but with with the adapted `75_375_dip` temperature cycle, which is displayed in Figure 4.23. The SGP30 sensor unit operated in parallel has ceased to operate properly, not producing a usable output. The measurement was interrupted multiple times for other uses of the hardware, maintenance or other issues correlated with the data acquisition.

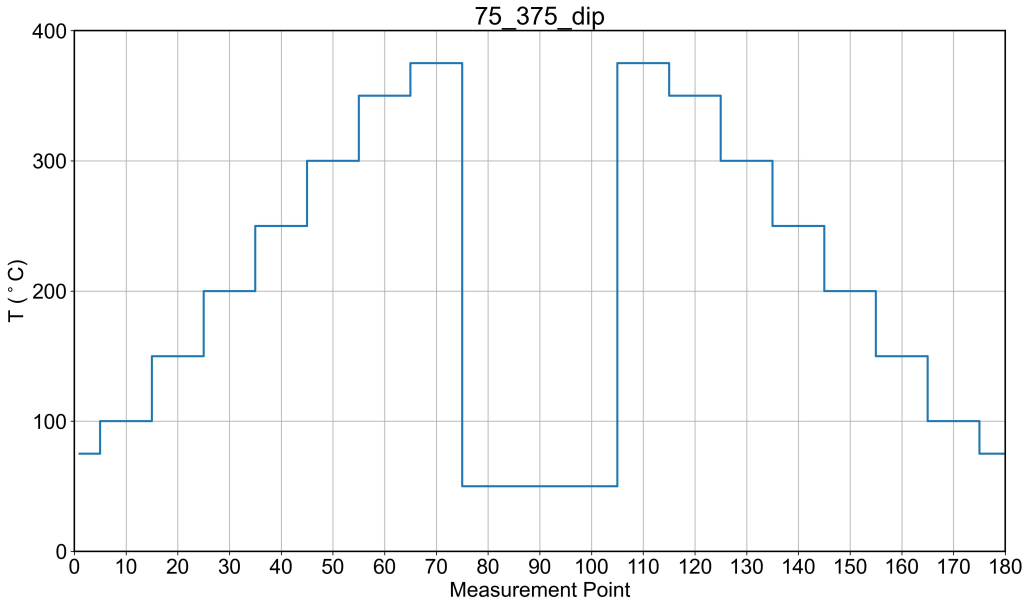


Figure 4.23: Time series of the operation temperature of the modulation 75_375_dip.

The ozone concentrations during this long term experiment are plotted in Figure 4.24. Periods of which no sensor data is available are highlighted by a red background. The best CNN architecture determined for this SGP40 has been trained again. Figure 4.25 shows the prediction of ozone in the test set, achieved through a split performed on shuffled and unshuffled data. Even the model trained on the random split across the whole time period produces a remarkably large statistical error while keeping the overall trend correct; the core of the predicted samples is scattered around the angular bisector. In the case of an unshuffled split, the MRE is much larger with a very significant amount of large concentrations being underpredicted.
The overall poor performance, especially compared to that of the same SGP40 sensor in the UBA data set, is likely caused by the unsuitable adaption of the temperature cycle with higher temperatures in general. As described and reasoned in Section 4.8, the weighted coefficients of the ridge regression (of an SGP30) have been the main motivation.

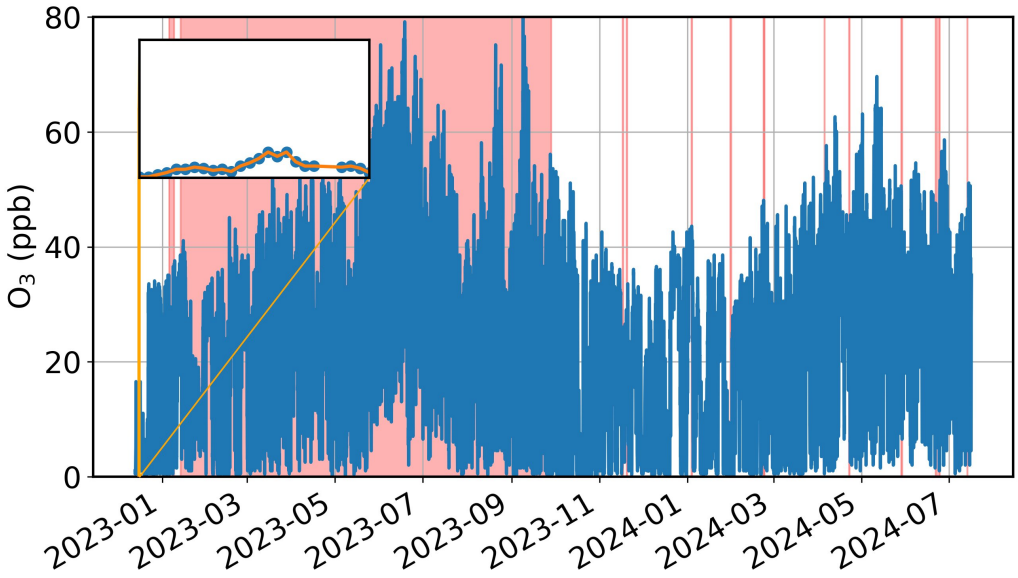
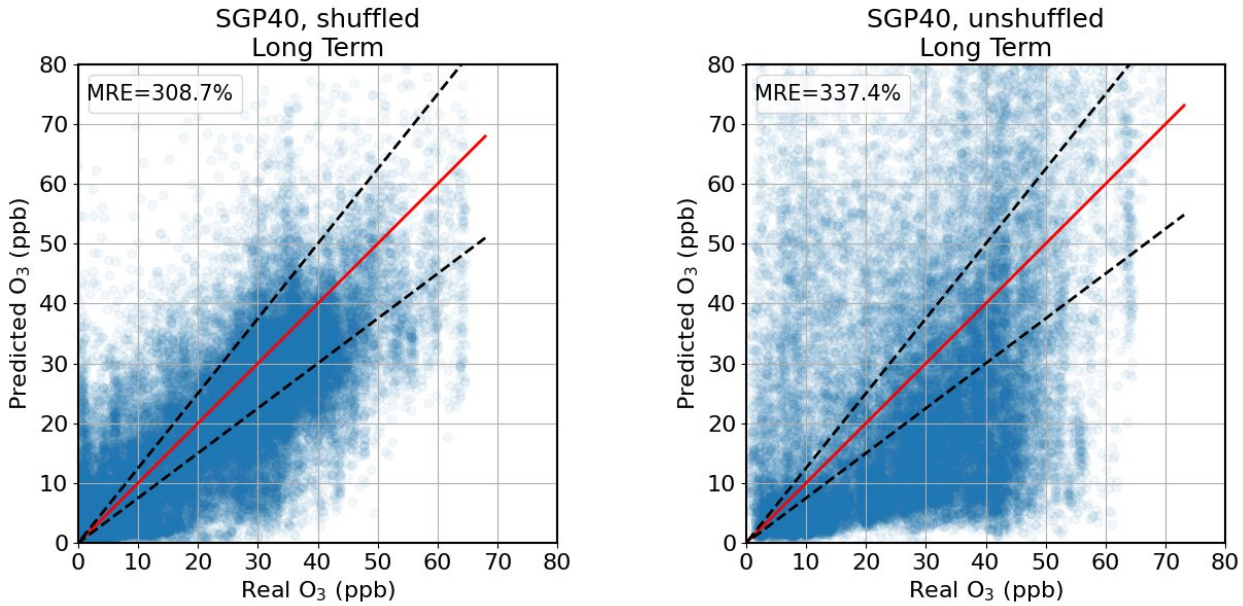


Figure 4.24: Time series of the ambient ozone concentration in Tübingen. Highlighted in red are the times of where the sensors experienced a downtime.



(a) (b)

Figure 4.25: Predicted vs. true ozone concentration based on an SGP40 and CNN.
(a) Shuffled data set split
(b) Unshuffled data set split

A smaller subset of the whole data, focusing on summer, is treated in the same fashion to isolate the influence of the poor modulation in comparison to the UBA experiment. This smaller subset from April to June, 2024 includes an equivalent amount of observations – a good 30 thousand. The resulting prediction performance is supplied in Figure 4.26. Ultimately, this poor performance mirrors that of the whole long term data set and underlines the unsuitable operation mode.
Though the differences of the shuffled and unshuffled split suggest an aging process, these results would be more significant when obtained on a well working measurement – i. e. with the „full“ temperature ramp. These results highlight the importance of low temperature steps for sensing and furthermore emphasize the complex dynamics of the whole measurement cycle. After all, more advanced analysis methods are necessary to investigate important temperature plateaus and fine tune the temperature ramps. The next paragraphs are devoted to different feature extraction schemes and SHAP analysis to provide further input.

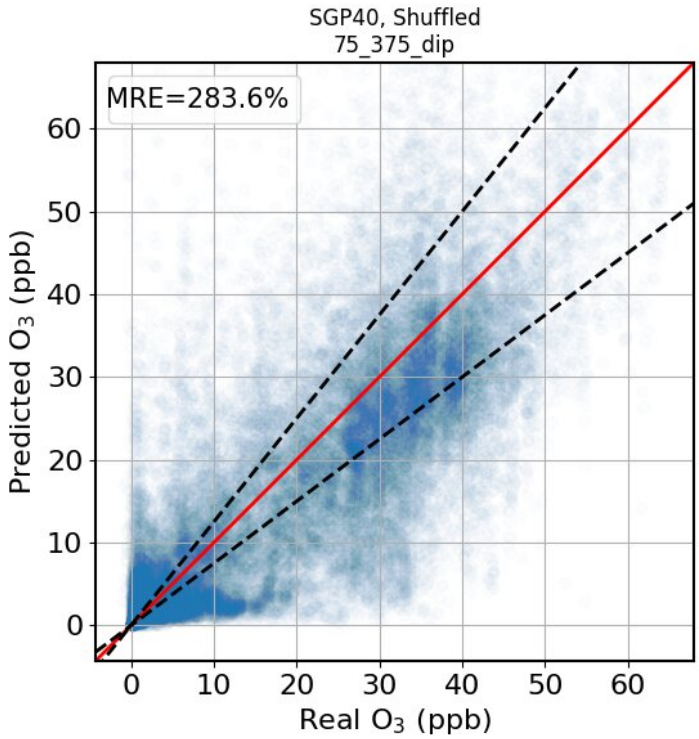


Figure 4.26: Predicted vs. true ozone concentration with an SGP40 in a smaller subset in Summer 2024 to isolate the effect of the temperature modulation 75_375_dip.

4.8 Feature Extraction – Identifying Temperatures of Interest

For a given architecture and number of targets, the number of trainable parameters in a CNN model scales linearly with the shape of the input data. As a consequence, a large number of independent training samples is required to build a robust model ingesting a great number of features. In case of the aforementioned CNN models – predicting ozone in the UBA data or pollutant gases in the GMS experiment based on an input of shape (180×4) – well above a million of those parameters need to be optimized. Figure 4.27 shows the linear relationship of trainable parameters dependent on the number of resistance measurements per temperature modulation cycle. It becomes obvious, that the model size scales directly with the number of input features. Two approaches to reduce the complexity of the input are directly conceivable. First, it is possible to keep the operation mode and therefore the chemistry and surface reactions during the sensing the same, while training the chemometric models on just a fraction of those measurement points. In the following, different approaches for the selection of important features will be tested and evaluated. Secondly, the modulation mode itself could be shortened, especially after it is understood which parts of temperature cycle are beneficial. However, the long term experiment with it's elevated start and end temperature has highlighted that selecting or tuning this temperature mode is challenging and features are not independent of each other. It is important to consider, that each point in time correlates to some extent with the others as is directly evident by the different shape and significance of the resistance response during heat-up and cool-down phase of the ramp used here. Clearly, more sophisticated methods of feature extraction than the mere interpretation of regression coefficients is needed.

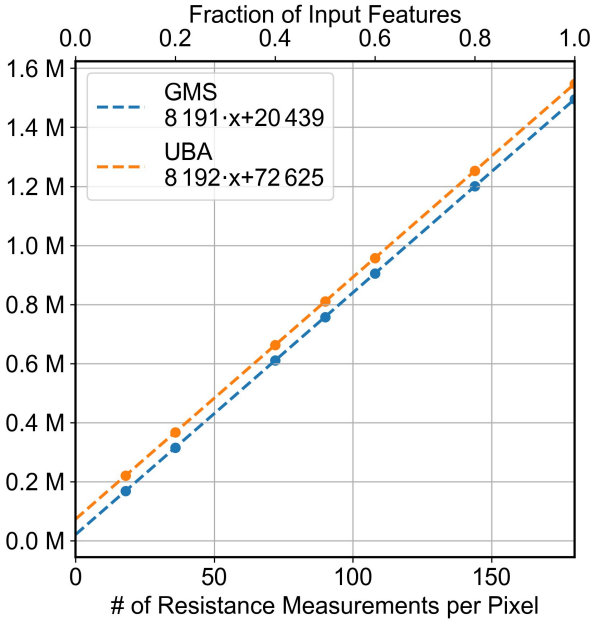


Figure 4.27: Number of trainable parameters for different shorter temperature cycles or fractions of randomly down-sampled features.

The simplest kind of feature selection that also serves as as a comparison or base line is the random selection of a certain number of input features. Figure 4.28 shows the MREs of prediction for CNN models trained on such randomly down-sampled fractions. For each fraction of input resistances tested, 20 repetitions of randomly selected `ramp_steps` – i. e. points in a single temperature cycle – have been chosen, disregarding all others in the $\mathbf{X}$ tensor. The average and standard deviation of the MRE in these sets are plotted. In contrast to the original feature space, the temporal properties of the transient response are not necessarily captured as intermittent resistances may be missing by chance. The consistently large error bars across the different fractions, in both the GMS and UBA data set, highlight the different significance of input features. The average MRE in the UBA data set approaches that of a CNN trained on all features – denoted by the dashed line – for increasing fractions and is almost equivalent at 80 % of the input data. For fractions larger than 40 % some random feature selections must yield an overall MRE lower than that of the reference as observed by the error bars extending below the dashed line. It is thus the goal of any model based feature selection, to meet these selection criteria and produce not only a slimmer but also more accurate model.
In the GMS data set on the other hand, the MRE seems to converge, but not close to the value of the full model. This finding suggests that the complete temperature ramp is indeed needed in this complex application with multiple gases to be quantified in mixtures. The shape of the resistance curve and transient response to all temperature steps is irreplaceable.

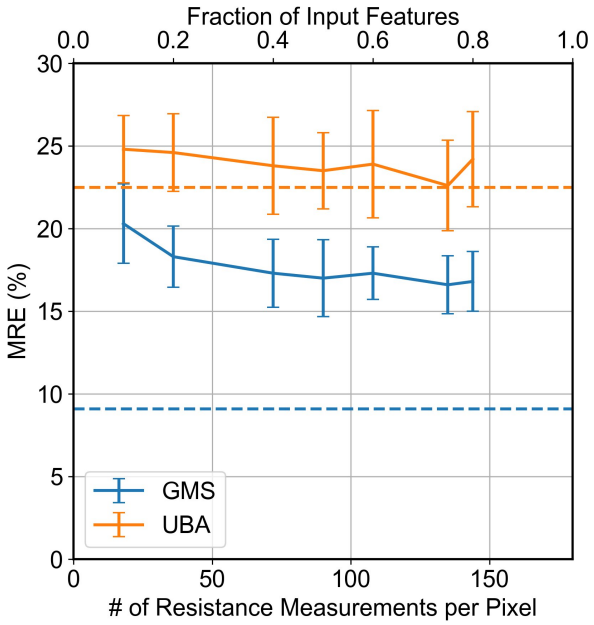
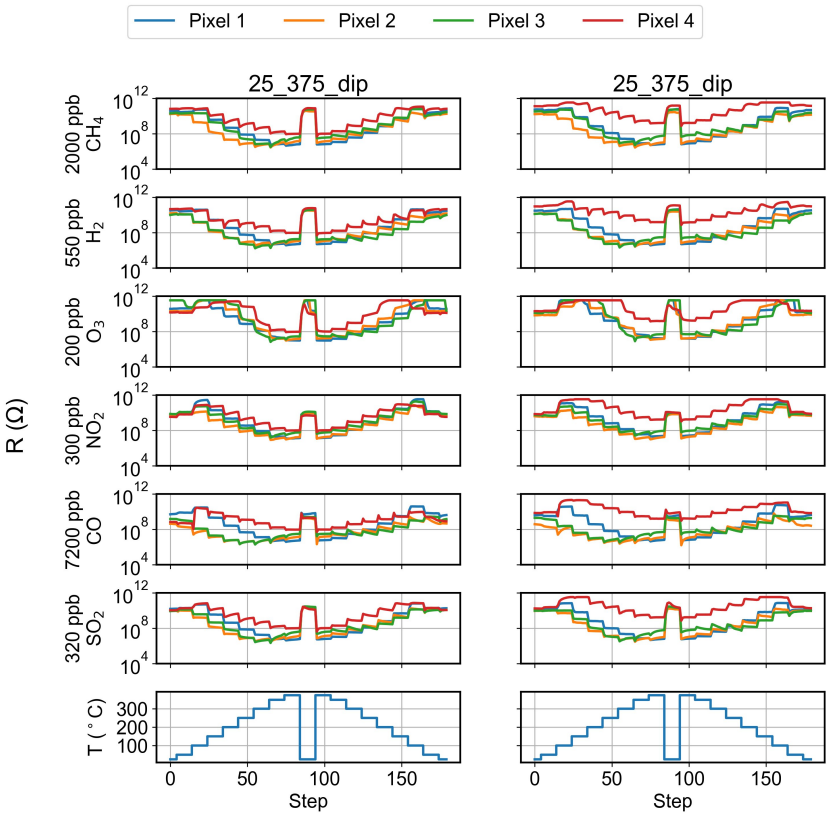


Figure 4.28: MRE in the test set for different fractions of randomly down-sampled features. For each fraction 20 random selections have been tested and averaged. The error bars reflect the standard deviation.

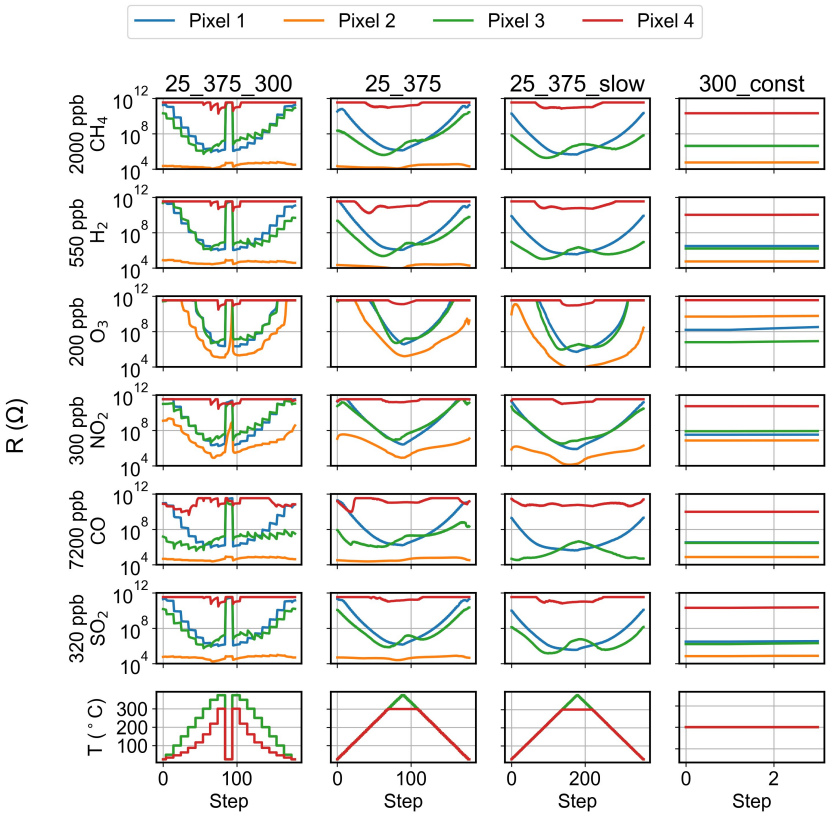
Dynamic Screening of Temperatures

Another classical approach is the investigation of single gas exposures, coupled with a sweep or scanning of operating temperatures. Yamazoe and Marui for instance report the sensor signal – though they call it sensitivity – of differently doped SnO_2 towards a number of analyte gases in dependence of temperature [134]. The change of resistance or sensor signal of CO for pristine SnO_2 is negligible for

all temperatures, but is severely enlarged for Pt doping below 100 °C. A similar scheme is followed in Figures 4.29, where the resistance is plotted for a medium concentration (relative to the GMS data set) of all gases when present individually in a background of 30 %RH. The temperature modulation cycle of two SGP30 units on the left is complemented by a set of SGP40 sensors featuring the same modulation mode as well as a fast and slow temperature sweep next to a constant reference. It should be noted, that the maximum temperature of pixel 4 of the SGP40s, which allow individual hotplate control, is 300 °C to avoid material degradation. This data set was acquired with Tim Hitzler during his Bachelor thesis [135]. All four pixels of the SGP30s show an overall similar resistance pattern – irregardless of the target gas – that is mostly governed by the temperature modulation with a stepwise decrease of the resistance with an increase of the temperature and vice versa. Deviations from this monotonous trend can be seen for most gases and pixels between room temperature and 100 °C. Comparing the two ideally identical SGP30 units, a significant difference in the absolute resistance of pixel four is evident, while the qualitative behavior is comparable. The SGP40 sensor operated with the same temperature cycle – rescaled for pixel 4, only – shows a similar resistance diagram. Most notable however, is the different shape of pixel 4 that cannot be explained by generally lower temperatures. Additionally, pixel 2 features remarkably low resistances especially for SO_2 and CO.



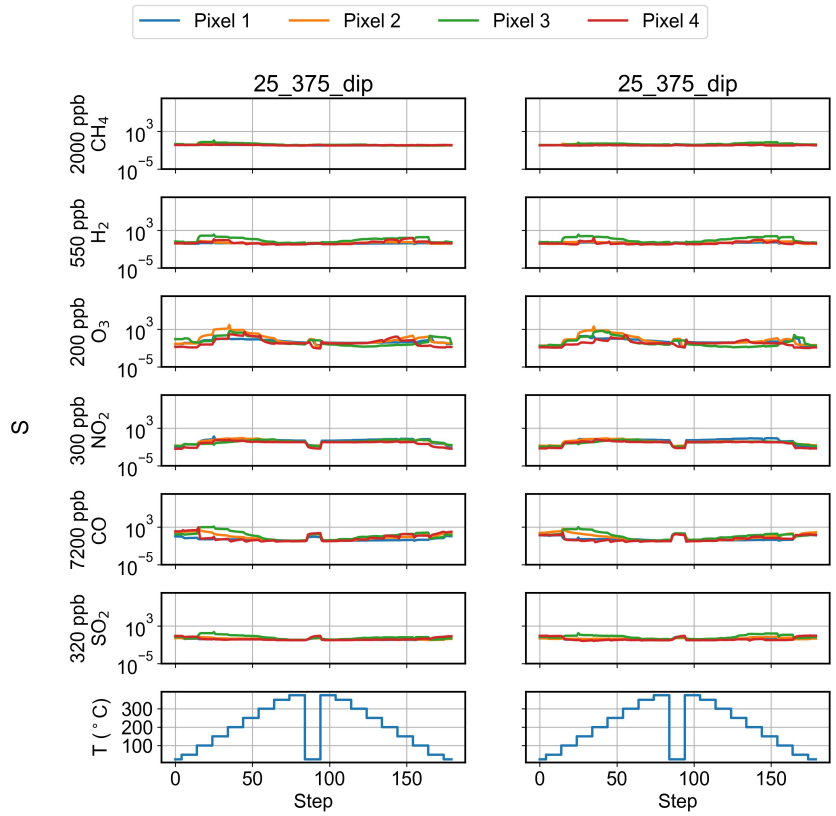
(a) SGP30



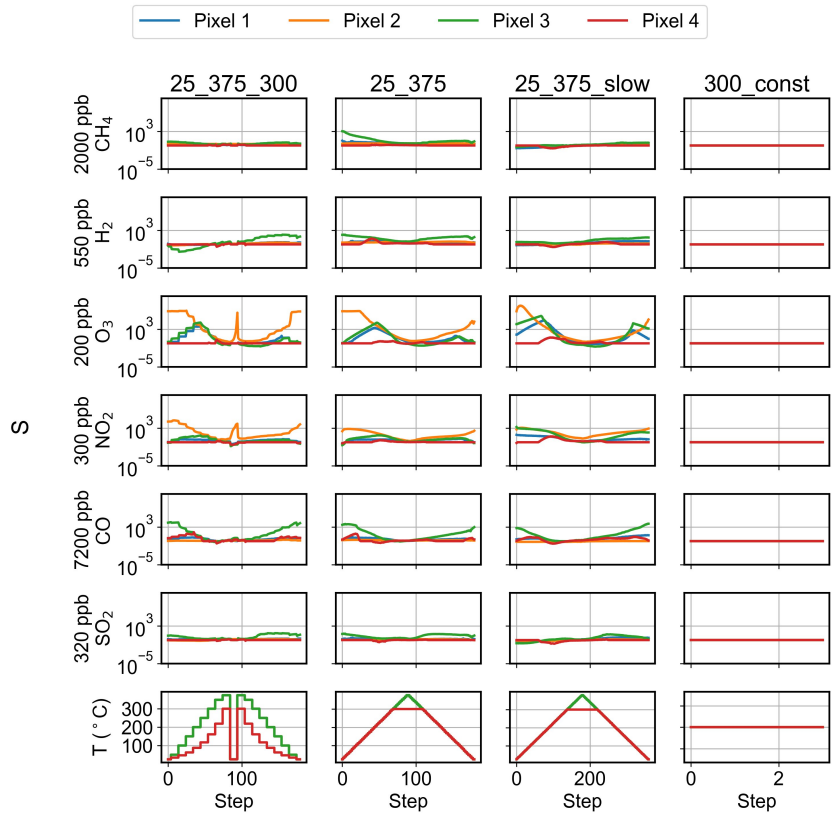
(b) SGP40

Figure 4.29: Resistance response during a single temperature cycle and exposure to single target gases in a medium concentration encountered in the GMS data set.

The same data is presented in Figure 4.30, but the resistance at any point is transformed to a sensor signal as defined by equation 2.1 with reference to a measurement cycle in a previous baseline, i. e. in the absence of the respective gas and in the same RH background. CH_4 produces little to no response in the SGP30. The response towards the seconds interferant gas H_2 in the GMS data set shows some positive signals of pixel 3 during the medium low temperatures not too dissimilar to those of SO_2. This could further explain, the inherent difficulty to predict SO_2 observed before. CO and O_3 produce the largest sensor signals for low and medium temperature steps, respectively, both during heat-up. Overall the similarity of all gases and pixels in striking.
The SGP40 operated in the same temperature mode on the other hand, produces much larger signals, especially for O_3 and pixel 2. But also pixel 1 and 2 have large signals during heat-up with a maximum around 150 °C. For the oxidizing gases O_3 and NO_2, the dip of pixel 2 is prominently featured. Apart from this dip, the finely grained triangular temperature sweeps produce very similar response curves. In all cases, there is a certain lack of discrete features giving further insight. Finally, this kind of analysis is limited to single gas and not directly applicable to mixtures or largely different concentrations.



(a) SGP30



(b) SGP40

Figure 4.30: Signal-like response during a single temperature cycle and exposure to single target gases in a medium concentration encountered in the GMS data set. The resistances are referenced to equivalent measurement points in the previous baseline condition as a sensor signal (equation 2.1).

SHAP Analysis

Shapley values present a pertinent figure to judge the importance of single sensors or sensing materials of an array for the prediction of target gases. A suitable example is the prediction of ozone in the UBA data set with a temperature modulated SGP30 and SGP40 sensor array. A ridge regression has been trained on flattened vectors of the size 720 containing all features of the four pixels. The resulting R^2 value in the test set is 0.771 and 0.761 for the SGP30 and SGP40, respectively. The Shapley values in Table 4.9 can now explain to what extent this is owed to the single sensing elements. For this, ridge regression models where trained on all permutations of the set of features from single pixels as described in Section 2.4. As Shapley values are additive, the sum of the individual contributions equals the R^2 of the model with all pixels. In the case of the UBA data and the SGP30, Pixel 4 followed by Pixel 2 are most important for the accurate prediction of ozone. Pixel 3 accounts for the least amount of the prediction performance. The dominant contribution of pixel 4 has already been exposed by the analysis of the ridge coefficients. The Shapley values of the SGP40 exhibit a comparable R^2 score with similar, but rather even contributions from the individual pixels.

Table 4.9: Shapley values for the different pixels of a temperature modulated SGP30 and SGP40 in terms of the R^2 score of a ridge regression predicting ozone.

Pixel	R^2 Contribution	
	SGP30	SGP40
1	0.155	0.163
2	0.211	0.218
3	0.166	0.218
4	0.240	0.245
All	0.771	0.761

The concept of Shapley values can be expanded and applied in the SHAP analysis, to make an analogous interpretations of features towards target predictions computationally feasible and most importantly comparable across different, complex chemometric models.
The following paragraphs reflect the SHAP analysis in [113]. Here, representative data points in the test set are used to evaluate and explain the inner workings of the three models trained for the GMS and UBA data set. For each of the target gases, four characteristic concentrations have been chosen: the minimum, maximum, and those that define the tertiles (33rd percentile) of the respective distribution. To account for variability and include statistics, the two data points with the closest concentrations above and below each of those values have been evaluated as well for the follow-up averaging. Listing 8 demonstrates, how the indices for these four times five values are extracted from the concentration $\mathbf{y}$ data. With these indices the associated $\mathbf{X}$ inputs are obtained.
For each of the selected objects of $\mathbf{X}$ from the training set, the SHAP values are calculated. The `LinearExplainer`, `TreeExplainer`, and `DeepExplainer` of the python package `shap` by Lundberg et al. have been used for the ridge regression, XGBoost, and CNN models respectively [32, 136]. Listing 9 shows the exemplary, iterative calculation for a CNN model. This requires the use of a representative background, here randomly sampled from the training set with $n = 1,000$.

```
num = 3
percentiles = np.arange(100//num, 101, 100//num)
percentile_indices = []

for p in percentiles:
    value_at_percentile = np.percentile(Ys, p)
    # get the closest value
    tmp = np.sort(Ys[Ys <= value_at_percentile])[-1]
    # get idx at sorted array of closest value
    idx = np.where(sorted(Ys) == tmp)[0][0]
    # get values around the percentile
    tmps = sorted(Ys)[idx-mult//2:idx+mult//2+1]
    # get all indices where these values are found
    idxs = np.where(np.logical_or.reduce([Ys == t for t in tmps]))[0]
    # idx = np.where(Ys == tmp)[0]
    percentile_indices.append(idxs)
# Add minimum as well
idx = np.where(sorted(Ys) == np.min(Ys[Ys >= 0]))[0][0]
# get values around the percentile
tmps = sorted(Ys)[idx:idx+mult]
# get all indices where these values are found
idxs = np.where(np.logical_or.reduce([Ys == t for t in tmps]))[0]
percentile_indices.insert(0, idxs)
idxs = np.array(percentile_indeces).reshape((num, 5)) # shape: (num+1) x 5
```

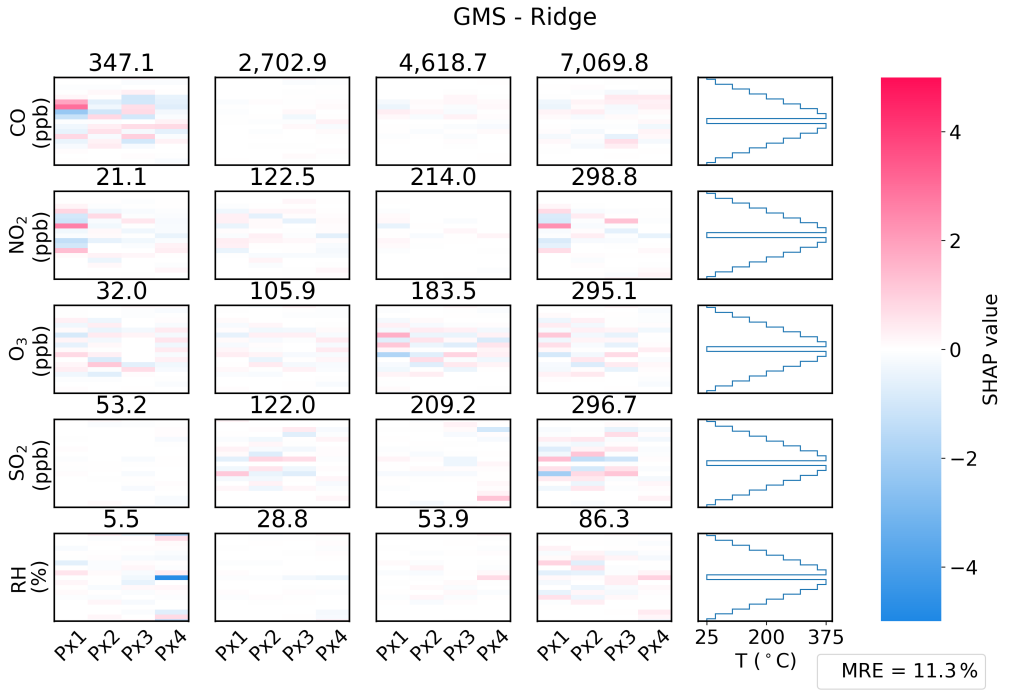
Listing 8: Code snippet to select the indices of five samples each around representative concentrations.

```
# create model background out of randomly sampled train set
background = Xs_train[np.random.choice(Xs_train.shape[0], 1000, replace=False)]

e = shap.DeepExplainer(model, background)

shap_values_all = []
Xs_shap_all = []
Ys_shap_all = []

for i in range(num):
    Xs_shap = Xs_val[[idx[i] for idx in idxs]]
    # plot mean as well
    Xs_shap_all.append(Xs_shap)

    shap_values_temp = e.shap_values(Xs_shap)
    shap_values_all.append(shap_values_temp)

shap_values_mean = np.mean(shap_values_all, axis=0)
Xs_shap_mean = np.mean(Xs_shap_all, axis=0)
# shape: 4 samples x 180s x 4 pixel x 1
```

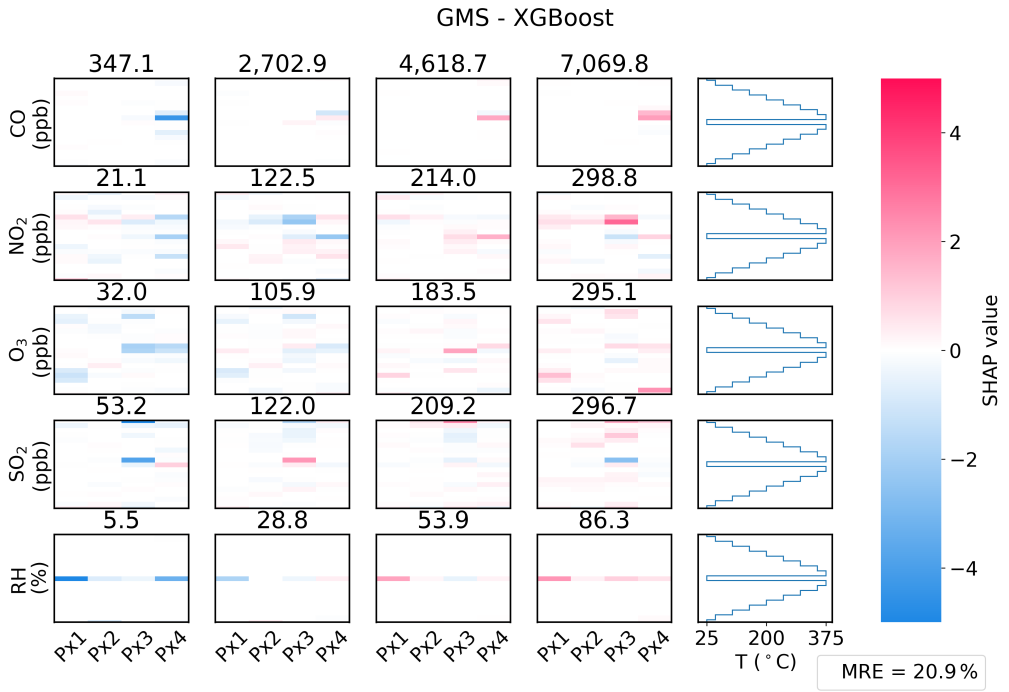
Listing 9: Code snippet to calculate SHAP values for a set of averaged data points defined by indices.

For each gas, four interesting concentration regimes have been chosen with five samples each. For final analysis, the SHAP values are averaged over those five closely related samples. An illustrative method of interpreting the obtained (averaged) SHAP values lies in color coded heat maps. Figure 4.31 and 4.32 present the heat maps for the three models in the GMS data set. Within each of these images, a column of rectangles corresponds to a single sensor pixel. The temperature cycle progresses from bottom to top as indicated by the staircase temperature ramp on the right. Each row of subplots presents one target gas with increasing concentration from left to right. To obtain information about the significance of single temperature steps, their ten SHAP values (five for the first and last step) are averaged.These images have a shape of (19×4). Besides its usefulness for the interpretation of singular temperatures, these findings could be used for the alteration or redesign of the temperature modulation mode, even specific for individual sensing elements in case of the SGP40. Just like Shapley values, SHAP values are additive and their sum for a single object/temperature cycle equals the predicted concentration or its deviation from the mean. Therefore their magnitude can greatly differ for some models. For instance, the absolute value for significant features is larger in models that extract only few of those features for the same predicted value. Similarly, models with conflicting information yield larger absolute values for compensation. Additionally, the magnitude naturally differs for different gases or applications, where the target values are of different orders of magnitude. Therefore, the SHAP values have been scaled by the reciprocal standard deviation per model and gas. These scaled mean SHAP values are displayed with a diverging color scale from blue to red.
Negative blue values highlight a feature (measured resistance at a certain time and temperature) that causes the model to predict a concentration that is lower than the mean of the concentration distribution. Accordingly, red color spots coincide with features that are responsible for a relatively high foreseen concentration. Consequently, a well working model should produce solely low or negative SHAP values for small concentration, leading to mainly blue colored spots on the left side of the figures. High concentrations on the other hand should be explained via red features. In Figures 4.33 and 4.34 that explain the O_3 prediction in the UBA experiment, this type of heat map is complemented by the raw, log-transformed resistance values in the upper row with the shape (180×4). The colorbar scale of the log-transformed resistances and the SHAP values is kept consistent over all six figures.

GMS



(a)



(b)

Figure 4.31: SHAP values for the pollutant prediction with
(a): Ridge Regression
(b): XGBoost.
For each target gas, the representative concentrations increase from left to right. The averaged and weighted SHAP values for each pixel (left to right) and temperature step in the modulation cycle (bottom to top) are visualized with a diverging blue-red color scale.
Reproduced from [113]

While relative humidity covers the lowest numerical range presented to the models, it produces the most pronounced SHAP values observable by the strongest primary colors in all three models. For the ridge regression for instance, the highest temperatures and the dip in the middle of the cycle produce signals most responsible for a prediction. These contributions though, are not at all homogeneous over the sensor's pixels or temperature steps. For the smallest concentrations, the dark blue patterns are mixed with red spots, which by themselves would lead to the prediction of a high values and in sum account for uncertainty or noise. The most meaningful features, that give rise to correct predictions over the whole concentration range, change from uniform dark blue to dark red from left to right. The most prominent example is the low temperature dip of pixel 4 for RH. For all other analytes, the contributions are mixed. The same pixel produces correct CO predictions at increasing temperatures. In general, pixels 4 and 2 appear to give the most accurate predictions with fewer contradicting contributions. As expected, the medium concentration values have lower associated absolute SHAP values. Pixels 1 and 3 at high temperature on the other hand, produce correct predictions of SO_2 and NO_2 most pronounced for the highest concentration. Overall, medium temperatures in the heat-up phase give decent predictions that are rather noisy and not too explicit. The fact, that the SHAP values are not mirrored by the dip in the middle in contrast to the operation temperature, underlines the complexity of the produced resistance patterns. The individual temperature steps are not at steady state, but depend strongly on heater sequence and history.

The XGBoost regressor in Figure 4.31(b) with somewhat shallow trees focuses on fewer features with inherently less contradiction. Again, the most pronounced values appear for RH, which is reflected by the lowest MRE per analyte. Again, the center of pixel 4 shows the largest significance, this time for all analytes but most pronounced for RH and CO. The same step of pixel 3 reveals meaningful information about ozone, but also the most obvious wrong explanation of SO_2 at high concentrations. The medium high temperatures, preferably during cool-down, contain most of the remaining extracted information.

The SHAP values of the CNN in Figure 4.32 resemble those of the ridge regression qualitatively. The successful quantification of RH due to the temperature plunge is the most consistent feature throughout the set. Yet, the CNN explains this not solely by pixel 4 but also the others. The found importance of this temperature dip is well in line with the proposed abrupt temperature changes by Schultealbert et al. [60]. In addition, the CNN has learned to extract resistances at the start and end of the cycle which correspond to another temperature well with a significantly smaller temperature change. This tendency to extract more features overall, is observed for all analytes. Besides the temperature dip and a certain preference of pixel 4, the medium to high temperatures are extracted and interpreted. Similar features of pixel 3 produce accurate predicted values for the concentrations of CO and O_3. NO_2 on the other hand is determined by pixel 2. Compared to the other models, much stronger implications are drawn for SO_2 which was the target gas with the lowest accuracy in general. Despite the obvious contradicting features extracted by the CNN model, it is able to interpret the data samples holistically in the best way, producing the lowest error metrics as summarized in Table 4.2.

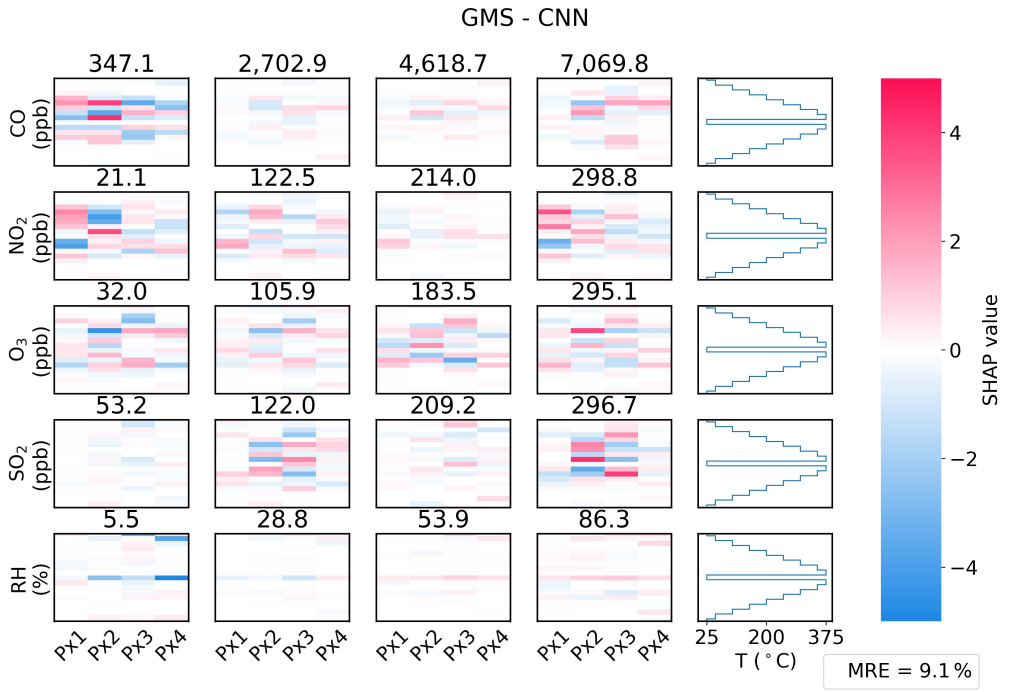
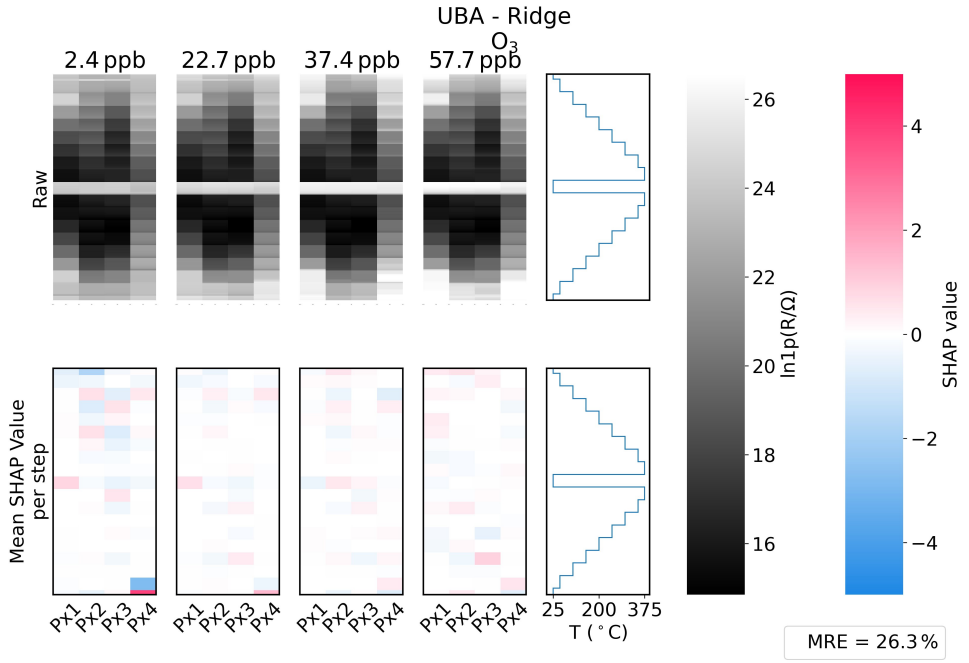
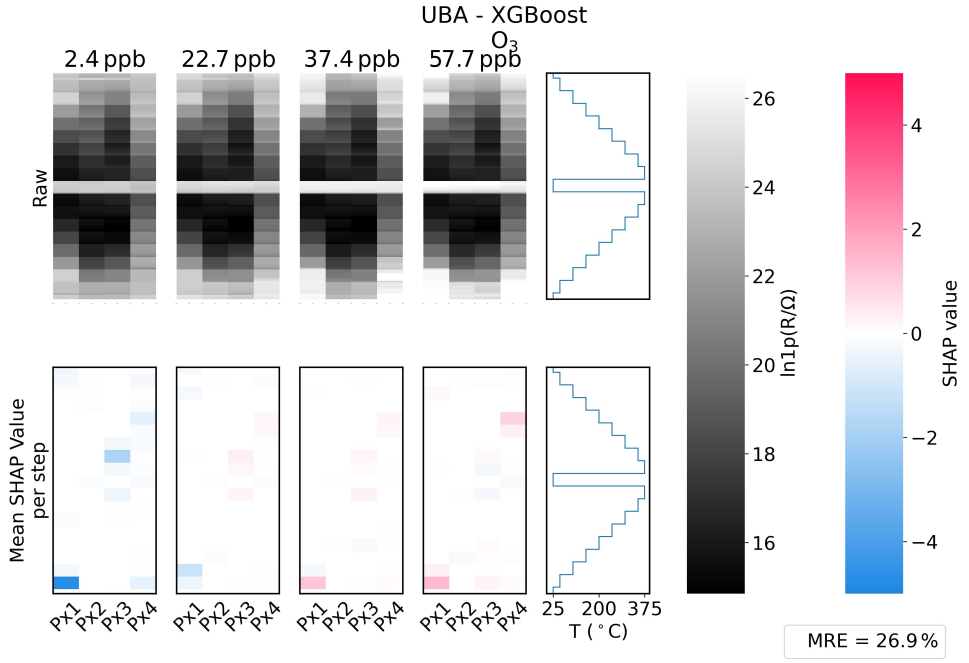


Figure 4.32: SHAP values for the pollutant prediction by a CNN.
For each target gas, the representative concentrations increase from left to right. The averaged and weighted SHAP values for each pixel (left to right) and temperature step in the modulation cycle (bottom to top) are visualized with a diverging blue-red color scale.
Reproduced from [113]

UBA



(a)



(b)

Figure 4.33: SHAP values for the O_3 prediction with
(a): Ridge Regression
(b): XGBoost.
The representative O_3 concentrations increase from left to right. In the top row, the logarithmized raw resistances are depicted as a grey scale image. The averaged and weighted SHAP values for each pixel (left to right) and temperature step in the modulation cycle (bottom to top) are visualized with a diverging blue-red color scale.
Reproduced from [113]

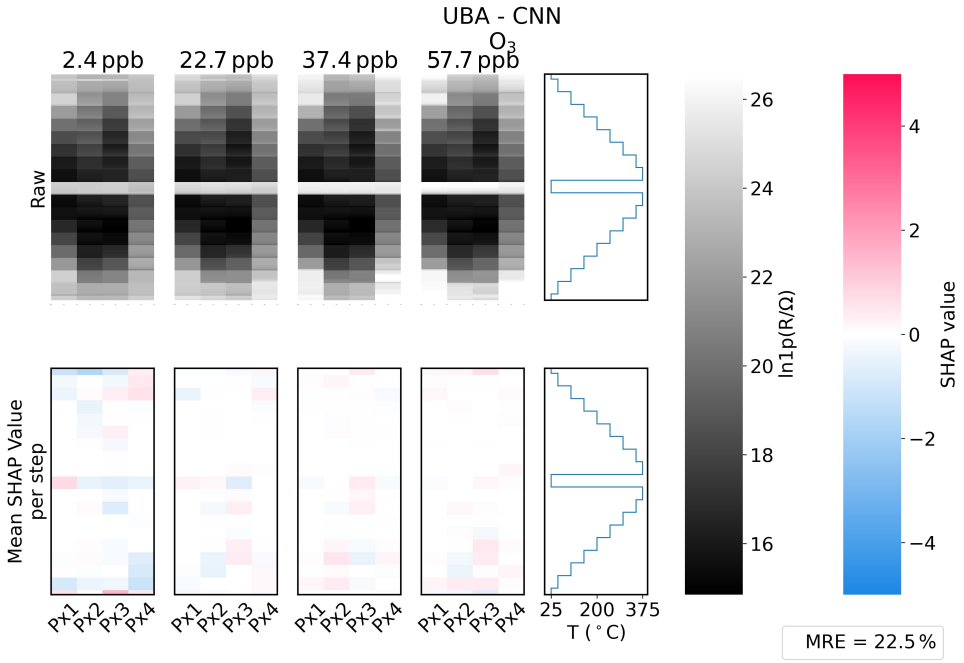


Figure 4.34: SHAP values for the O_3 prediction by a CNN.
The representative O_3 concentrations increase from left to right. In the top row, the logarithmized raw resistances are depicted as a grey scale image. The averaged and weighted SHAP values for each pixel (left to right) and temperature step in the modulation cycle (bottom to top) are visualized with a diverging blue-red color scale.
Reproduced from [113]

Comparing the raw data across the two data sets, lower absolute resistances are encountered in real outdoor air. Besides other reasons, this could simply be due to lower average ambient temperatures without the heat-up of a measurement chamber and plenty of air circulation.
The ridge regression produces its output based on very contradicting features, mostly from the cool-down phase with correct contributions of the last step, mainly from pixel 4. The importance of pixel 4 followed by pixel 2 has already been hinted at by the classic Shapley analysis, where the 180 features of the individual sensing elements have been evaluated as one.
In the case of the XGBoost regressor, only very few features have been focused on again. It extracts mostly Pixel 1 at the last, lowest temperature and a single prominent temperature step each, of pixel 3 and 4 during heat-up.
The superiority of the CNN becomes very clear in Figure 4.34. Not only are there few noisy, diverging feature effects, but overall there a many more different regions of the modulation cycle having a positive effect. The early heat-up steps and last low temperature are joined by useful information in the temperature gap. Some of these low temperature features, however, introduce noise. This is especially evident by the start of pixel 4 and the dip of pixel 1 for the lowest concentration. This temperature gap or dip creates the harshest change of temperature experienced in this modulation mode. Its importance has been reported by related publications [23, 60]. Partially, it has already been observed by Sears et al. who have cycled the heater voltage in a sinusoidal manner with different peak voltages. Different distinct features could be made out for higher amplitudes which appeared otherwise convoluted. However, the convoluted feature of lower voltages is larger in magnitude. Furthermore, an analysis of the sensitivity in dependence of the cycling frequency yielded no monotonous trend.

Consequently, there is no absolute maxim for the construction of temperature cycles and they have to be tested empirically and interpreted in hindsight; for instance by means of a SHAP analysis like this. Here, all models learned to extract different sets of features for both data sets in respect to ozone from the same temperature modulation. High temperature seems to influence the models' performance positively in the GMS data set, whereas contributions from low temperature steps are required in the real air outside in static conditions. These findings are supported to different extent by all three models. While this exhaustive temperature ramp produces useful features for both investigated application scenarios, the SHAP analysis could be used to tailor optimized operation modes for a given use case, for instance with fewer features and thus higher temporal resolution. The parallels between the three models suggest that small light weight models may be used as a proxy to assess the general applicability of a certain temperature cycle with expected accuracy gains by more complex ones like the CNN. Still, it is strictly necessary to create training data sets and optimize chemometric models for the precise targeted application. It also shows, that models trained for one scenario hardly succeed in another. For the following investigations, CNN models will be considered with the respective set of tuned hyperparameters where applicable.

SHAP Correlation with Analytes

A more numeric approach to interpret SHAP values without bias of chemical intuition, color scales, or a small number of samples is the correlation of the SHAP values with the true concentrations. A large positive correlation coefficient for a given sensing material and temperature step describes what was more intuitively represented by „correct“ contributions in the red-blue color scale in the figures of the previous section. After all, one would like the (averaged) SHAP values of a temperature step to be positive for high concentrations and negative for low concentrations for any given analyte gas. Figure 4.35 shows the Pearson correlation coefficients ρ of the analyte concentrations and the averaged SHAP values for each temperature step in the two data sets. The SHAP values in the respective test set have been averaged over all four sensing materials of the SGP30 on any given temperature plateau, because these temperature steps might be the smallest „building block“ when choosing a new modulation cycle. The upper panel shows the correlation coefficients for the individual gases in the GMS data set as well as the aggregated mean thereof. ρ_{RH} reflects precisely what has been seen in the visual inspection of the representative samples in Figure 4.32: The „best“ contribution stems from the temperature dip in the middle of the cycle and its end. The previously observed „wrong“ contribution of the beginning the cycle can be seen as a negative correlation. In general, a strikingly large number of temperature steps yield insignificant or even negative correlation. In parts, this is owed to neglecting the diversity of pixels by averaging, which have been seen to behave differently. Overall, all gases are met with some features producing significantly correlated sensor responses. Only SO_2 does not surpass the moderate correlation of $\rho = 0.5$, resulting in the highest errors of prediction observed before. The correlation with NO_2 concentration is also rather low, focusing on the heat up phase of the cycle. The average of the correlation per temperature step of all gases is shown in the lowest row of the first panel.
The correlation coefficients of the ozone concentration in the UBA experiment with the respective SHAP values largely reflect the findings in the visual interpretation thereof. However, by the averaging over all four sensing materials, much more significance is attributed to the features at (medium)

high temperatures. The low temperature steps at the beginning and end of the temperature cycle produces the largest SHAP values but were also subject to noise.
Those 90 features, producing the highest (average) correlation per data set have been selected for training slimmer CNN models, otherwise matching the architecture with respectively tuned hyperparameters.

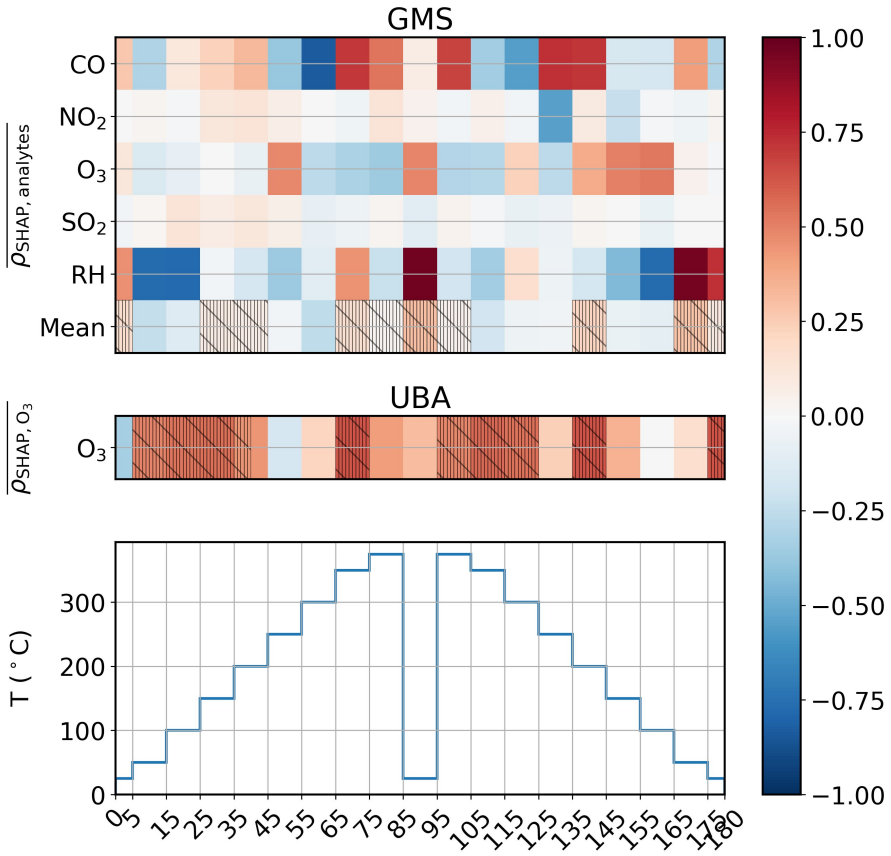
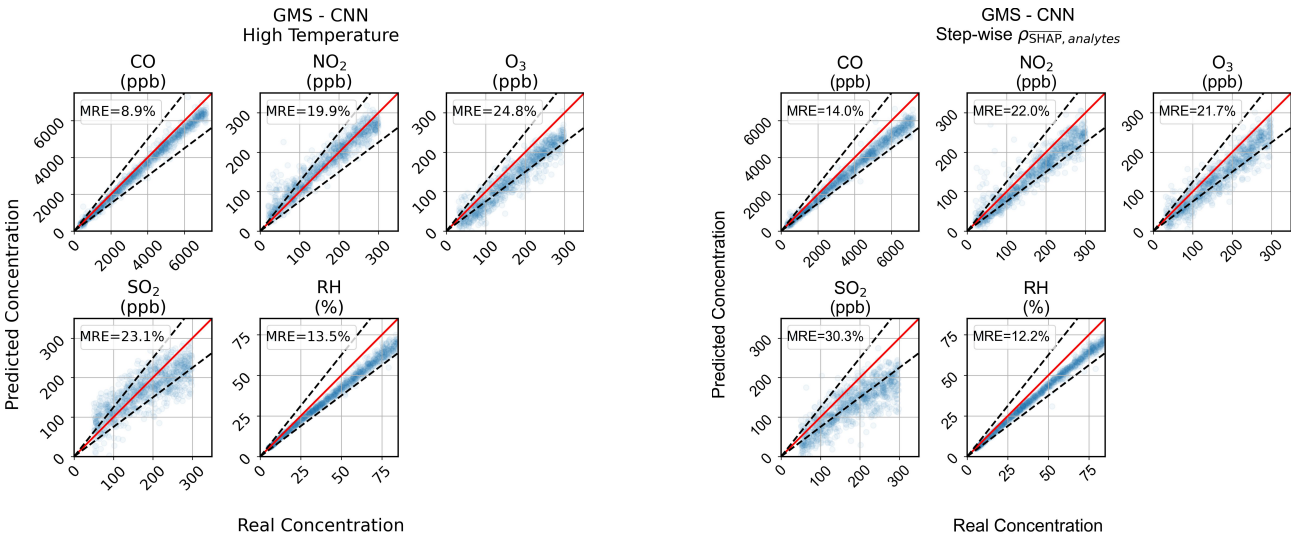


Figure 4.35: Correlation of SHAP values with analyte concentration averaged per temperature step all SGP30's pixels.
Hatched features/temperature steps are chosen for training a slimmer CNN model.

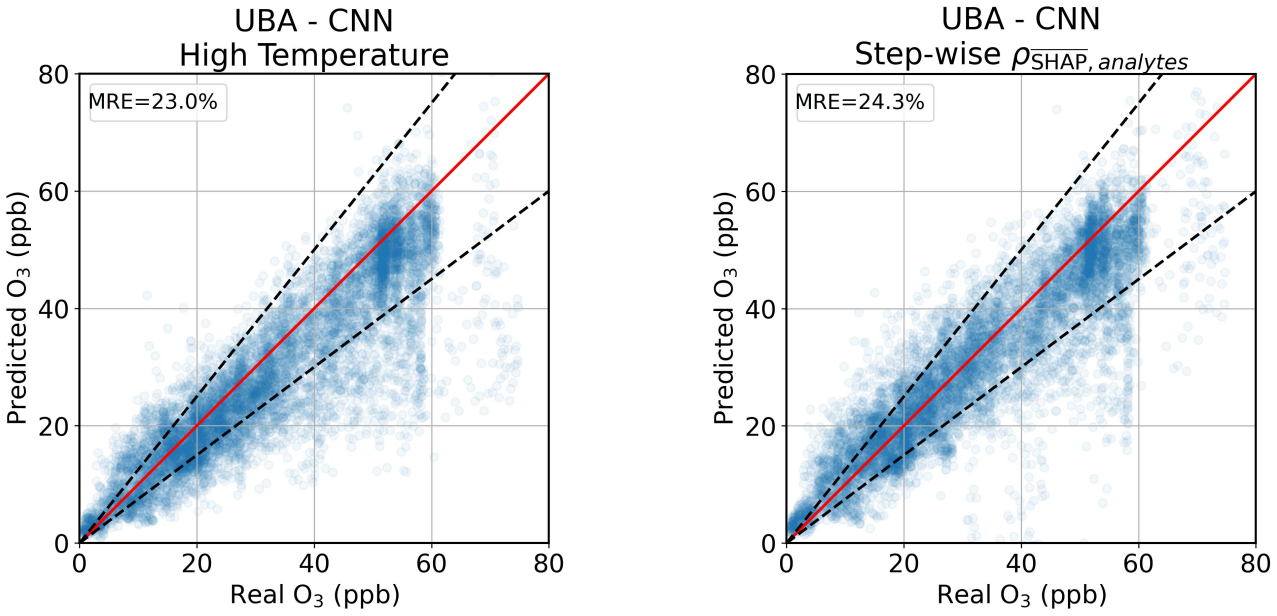
Figure 4.36 shows the predicted vs. true plots for CNN models trained on the 150 high-temperature features – selected by the ridge regression coefficients in the UBA set – and the 90 features selected by the SHAP and correlation analysis. The averaged MRE is 18 % for the high-temperature features and 20 % for those 90 selected by the SHAP analysis. Both of these results are well comparable with the randomly selected features in Figure 4.28 yielding between 15 and 20 %. The numerically better accuracy of the high-temperature features can mainly be observed for CO and O_3, but even the detection of RH and especially SO_2 are hampered. Similar to those selected from the SHAP values, high concentrations are systematically underpredricted. In line with the random feature selection, no selective feature extraction conserving the prediction performance can be reported.



(a) (b)

Figure 4.36: True vs. predicted pollutant concentrations with CNN models trained on a subset of features
(a) 150 high temperature features, including the „dip“
(b) 90 step-wise features with the highest correlation of averaged SHAP values and actual analyte concentration.

Figure 4.37 shows the results of the same feature extraction scheme for the UBA data set. Here, the respective features with the highest SHAP value correlation with ozone have been used in addition to the same high-temperature steps. In contrast to the GMS data set, the general trend and accuracy is maintained in both cases, no obvious systematic errors occur. The more severely truncated set of features selected due to the SHAP analysis produces a slightly higher relative error. Still, the high temperature features preserve a lot of the information, much in contrast to the poor performance of the elevated temperature cycle in the long term experiment. These results highlight that the individual temperature steps are not at all independent of each other but have a chemical memory of the sensor's operation temperature history. Interestingly, it is sufficient to operate the sensor with a low temperature start that is surrounded by the stepwise decrease and increase thereof. It is not strictly necessary to record or interpret the resulting resistances as long as the following resistance response at medium low temperature is considered. Moreover, the abrupt temperature dip, which has been preserved and broadened in the `75_375_dip` temperature cycle, apparently does not provide this information. While it's obvious that a single temperature step cannot be interpreted individually, it appears logical to fully define or characterize a single step or building block by multiple attributes. Besides the explicit factors like residence time and temperature these could include a temperature gradient (compared to the previous step) as well as the previous minium and maximum temperatures. A more systematic approach applying and comparing those elementary units in defined (laboratory) conditions would be needed.



(a) (b)

Figure 4.37: True vs. predicted O_3 concentrations with CNN models trained on a subset of features
(a) 150 high temperature features, including the „dip“
(b) 90 step-wise features with the highest correlation of averaged SHAP values and actual ozone concentration

SGP40

The SHAP analysis in Figure 4.38 reveals different contributions of the SGP40's particular sensing elements compared to its predecessor, as was already recognized by the Shapley analysis. Pixel 4's strong impact is diminished, which cannot only be explained by the introduced capping of the temperature. It also exhibits rather high resistances, quickly out of range for the low temperature steps. Instead, pixels 2 and 3 dominate the prediction with most prominent features at medium lower temperatures. Still, conflicting information can be observed. Overall, the same low temperature regimes are deemed important for an accurate prediction, even with changed or reordered sensing materials. Once more, this explains the inadequate quantification in the long term experiment, where the lowest temperatures have been raised or omitted.

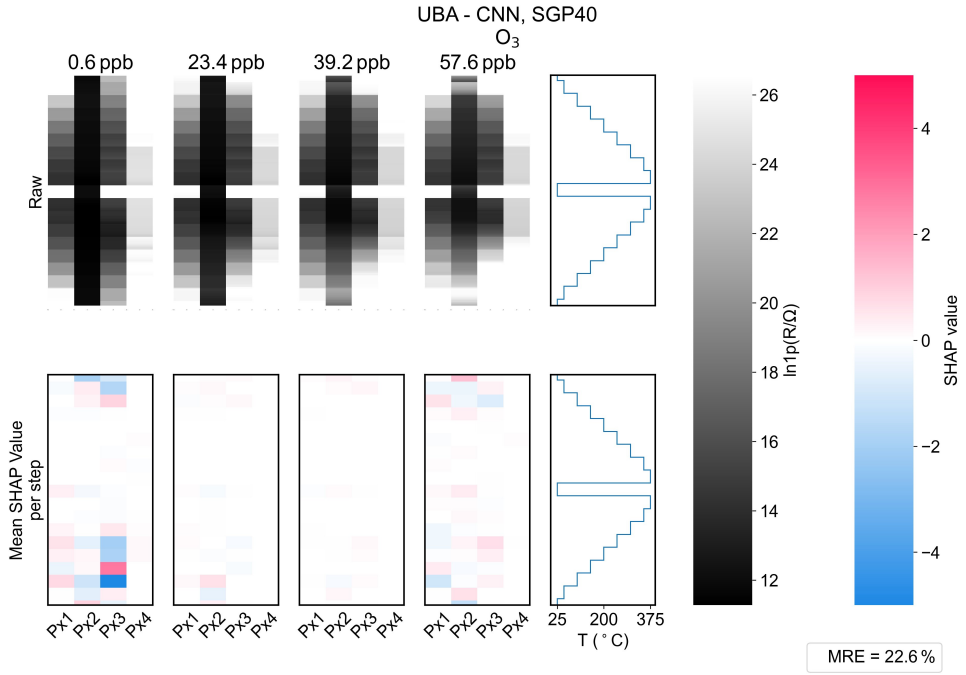


Figure 4.38: SHAP values for the O_3 prediction by a CNN with an SGP40.
The representative O_3 concentrations increase from left to right. In the top row, the logarithmized raw resistances are depicted as a grey scale image. The averaged and weighted SHAP values for each pixel (left to right) and temperature step in the modulation cycle (bottom to top) are visualized with a diverging blue-red color scale.

4.9 Atmospheric Ozone Prediction – Other Sensors

The SHAP analysis paves the way for interpreting and understanding the importance of input features not only in respect to the operation temperature but also the choice of sensors or sensing materials. This work has brought up the SenPi sensor platform to operate analog sensors in the same fashion and screen new materials for their sensing capability. Further, the development of the standalone eNose Platform facilitates the deployment of sensor units at different places of interest and especially in vicinity of reference stations. Here, the SHAP analysis can be used to validate and compare the sensing process across different experiments. The following paragraphs summarize the findings and results obtained with these novel sensor instruments and compare them to the results obtained with CNN models in the UBA data set.

SenPi

The SenPi instrument described in Section 3.2.2 has been developed to ease the testing process of newly synthesized sensing materials – as well as analog sensors in general – in a similar fashion to the commercial sensors in the previous experiments, relying on temperature modulation. As such, one unit equipped with an in-house SFO sensor and a commercial TGS2600 operated with the `74_425_dip` temperature cycle defined in Section 3.2.2 has been deployed alongside the eNose Platform starting on October 11th, 2024. This period marks the decline of daily ozone concentrations with the progress of fall and winter, as observable in the overview in Figure 4.42.

SenPi Calibration

Even when buying components with high precision and low tolerance, electric readings and outputs are object to error. Depending on the circuit, these errors might add up, propagate, and even get amplified. Therefore, thorough calibration is often necessary for measurement equipment like the SenPi. Figure 4.39 shows the calibration curves for the two analog sensing ports of four units. The analog output voltage of the logamp is inversely proportional to the logarithm of the resistance under test, as described by equation 3.2. 18 known resistances spanning a range of 100 Ω to 100 MΩ have been measured. It can be seen, that low resistances below 1 kΩ lead to measured voltages exceeding the trend line. These values have been masked for the optimization of the fit parameters. Table 4.10 summarized these fit parameters in comparison to the ideal values of SenPi unit `e4a1`, which has been used in the following ozone prediction. The excellent R^2 value for all units and ports validates equation 3.2. Moreover, the fit values are generally in good agreement with the ideal component values. The highest deviations, up to 25 %, are observed for the „reference“ resistor. These fit values are used in the measurement of resistances by the SenPi unit prior to analysis.

Table 4.10: Ideal and fitted parameters of the LOG114 output used for the readout of the analog sensors of SenPi unit e4a1.

Parameter	Ideal	Fit Port1	Fit Port2
a – S_{factor}	0.665	0.659	0.662
b – $\dfrac{R_6}{R_5 + R_6}$	0.760	0.767	0.761
c – R_{ref}/Ω	1.62×10^6	1.194×10^6	1.555×10^6
d – V_{ref}/V	2.500	2.540	2.506

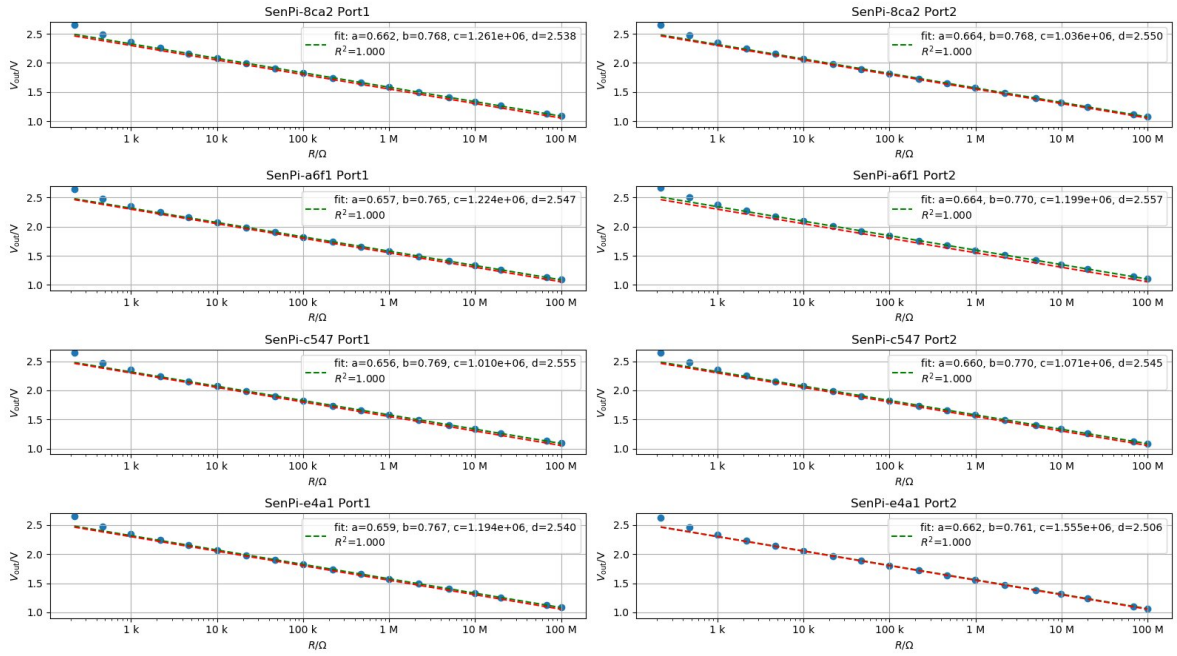
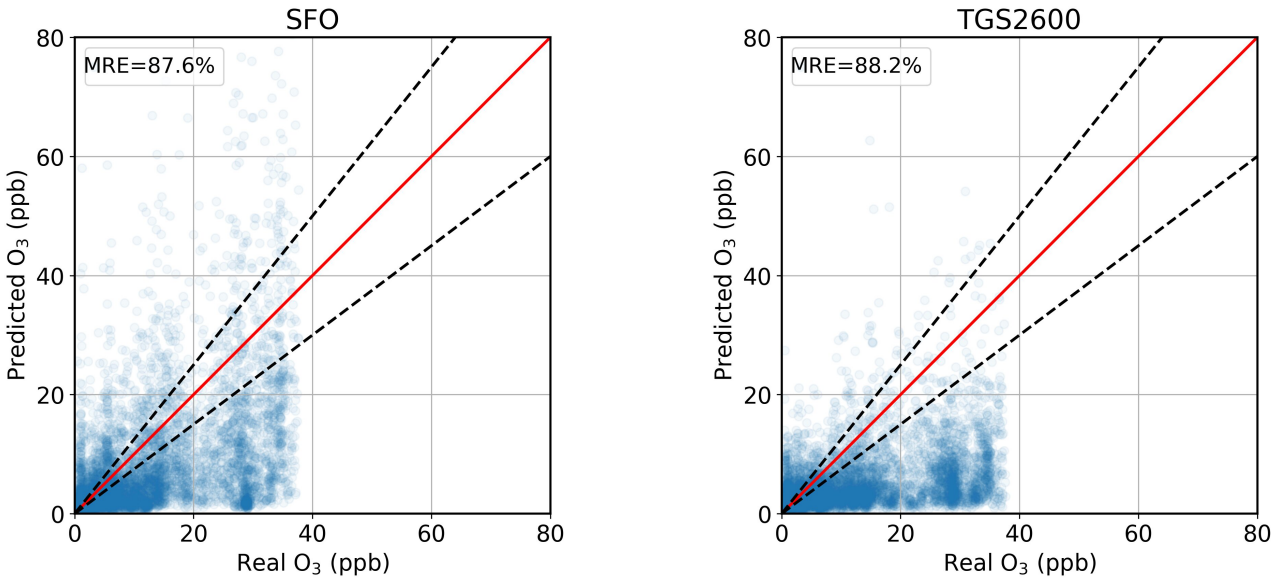


Figure 4.39: Calibration curves of the SenPi modules.

Ozone Prediction

Again, the same procedure as in the UBA experiment has been followed for training a CNN to predict ambient ozone. Since new materials and a different excitation are used, the hyperparameters have been tuned once more in a grid search. Figure 4.40 shows the established prediction plots for the best CNN model, respectively. The sobering performance of both sensors is evident by the large MRE and the overall spread or scattering of the sample points. A look at the ozone concentrations of the last quarter of the year reveals that the results are not too surprising as the majority of the data corresponds to ozone concentrations around or even well below the estimated LODs in the previous endeavors. Even though these values were established with different sensor units – namely SGP30 and SGP40 – the LODs of the SFO and TGS2600 can be assumed to be in a similar order of magnitude in a best case scenario or higher due to the single sensing material. Both sensors or materials – the TGS2600 is based on SnO_2 [137] – are not explicitly known or chosen for their exceptional ozone sensitivity. Instead, the TGS2600 is a well known and established sensors, historically in use as a general purpose reference sensor in the author's work group that has been replaced by an SGP40 (see Section 3.2.1.

SFO on the other hand features excellent sensitivity towards another oxidizing gas, namely NO_2 [138]. Additionally, the temperature/voltage cycle features fewer distinct steps with less abrupt temperature changes – which are found to be important – due to the added thermal mass.

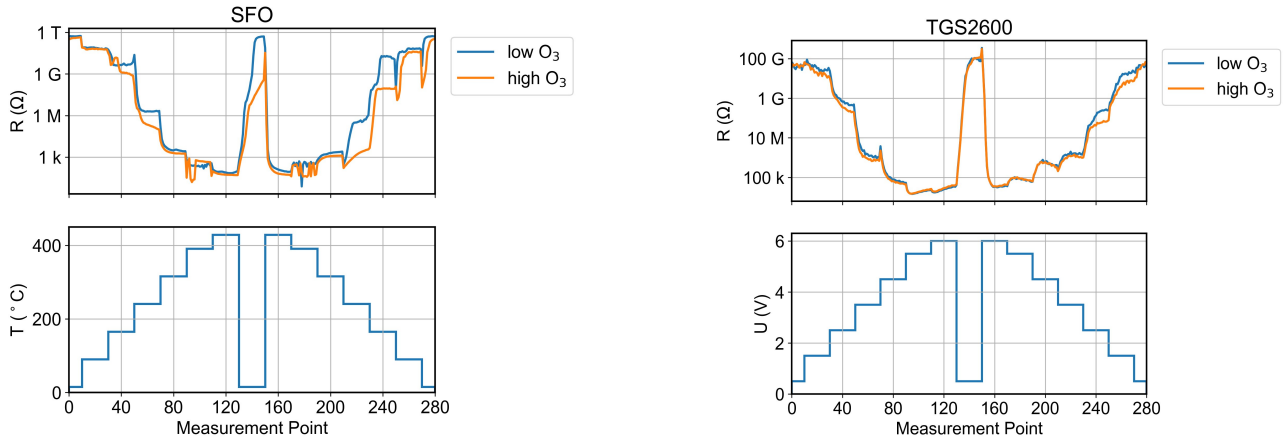


(a) (b)

Figure 4.40: True vs. predicted O_3 concentration with a CNN trained on a temperature modulated
(a) SFO sensor
(b) TGS2600.

Figure 4.41 shows two exemplary temperature cycles for a low and high concentration of ozone for both analog sensors. The same voltage modulation as reasoned by the temperature calibration of the SFO sensor has been applied to the TGS2600. The resulting temperature range is much narrower which can be observed in the smaller overall spread of resistances compared to the SFO sensor. The latter exhibits very high resistances well outside the calibration range for the two to three lowest temperatures. The highest temperature on the other hand, results in resistances well below 1 kΩ that can not be measured accurately anymore, giving a noisy response. For both sensors, the low and medium temperature or voltage steps produce the largest difference in the resistance pattern of the two extreme ozone concentrations.

While the SenPi apparatus has not produced an easy to predict body of data, the device has been proven to work for the intended use. In detail, small analog sensors – produced in-house or of commercial nature – can be operated over a wide range of thermal excitation with a parallel readout of the sensors' resistance spread over several orders of magnitude. The missing accuracy for ozone prediction can be mainly explained by the winter season with concentrations that are too low. Further, the temperature modulation has not been optimized.



(a) (b)

Figure 4.41: Exemplary resistances/features for a low and high concentration of O_3
(a) SFO sensor
(b) TGS2600.

eNose

The previous positive results about the applicability of temperature modulated sensors, namely the SGP40, for the prediction of pollutants in real ambient air have resulted in the development of the modular sampling and measuring device described in Section 3.2.3. Such an eNose Platform device has been put to test in the second semester of 2024 in Tübingen with the procedure established in the UBA experiment. Figure 4.42 shows the evolution of the ozone concentration for training and testing of the apparatus.

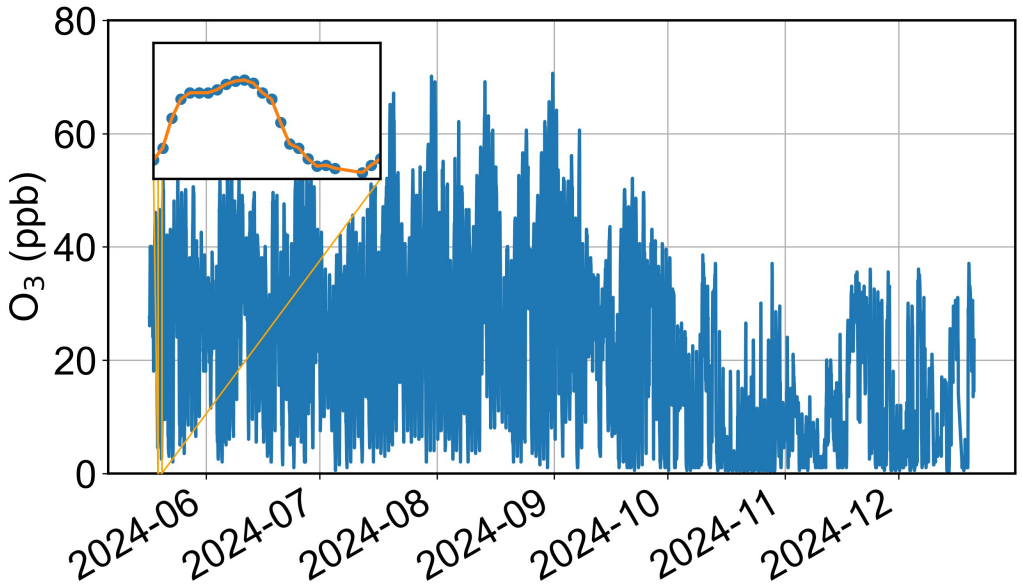
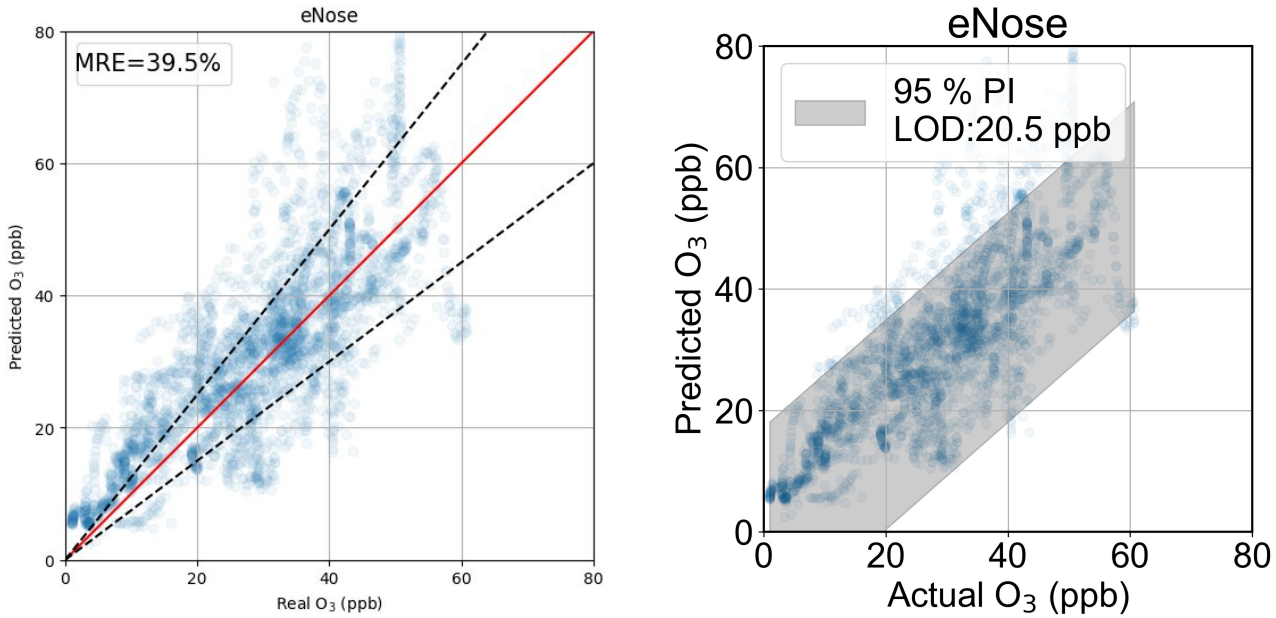


Figure 4.42: Time series of the ambient ozone concentration in the second semester of 2024 in Tübingen.

The best CNN architecture as determined for the prediction of ozone in the UBA with an SGP40 has been trained with the eNose Platform data. The resulting scatter plot of predicted vs. actual concentrations in the test set is given in Figure 4.9 and complemented with the estimated LOD. Both, the MRE and LOD are significantly higher than for a bare SGP40 exposed to the ambient. The largest difference between both scenarios is not only the use of a sensor enclosure, but the additional sampling procedure. Not only does the preceding sampling reduce the overall measurement frequency and thus number of training samples, it also limits the volume of the gas available to the sensor. In summary, the measuring scheme with the new sensor platform is validated, in an even longer period than the original UBA experiment. Yet, the added complexity calls for additional optimization and testing in terms of the whole operation. After all, errors may propagate and the majority occurs in the sampling process [139]. Still, the proposed unit already presents a capable analytical instrument and the whole analytical process has successfully been demonstrated.



(a) (b)

Figure 4.43: Predicted vs. true ozone concentration based on an SGP40 and CNN.
(a) with 25 % error boundaries
(b) with 95 % prediction interval and estimated LOD

4.10 Air Quality in Trento

A second unit of the latest eNose Platform platform has been deployed in „Parco Santa Chiara a Trento“ directly next to a public measurement station in Trento, Italy. The very close position of the sampling instrument gives access to highly accurate reference readings. Similar to the approach in Tübingen, the hourly reference data supplied by an API, here from the EEA, has been interpolated and used as the ground truth. In addition to O_3 NO_2 and SO_2 measurements are also available. The time series of the three pollutants for the duration of measurement and evaluation is shown in Figure 4.44. Generally, the observed O_3 concentrations are in the same range as in the UBA data set in Tübingen, up to 50 ppb. However, the highest concentrations are reported in the beginning of the data set, before the middle of October, corresponding to late summer (and the beginning of fall) with higher ambient temperatures and irradiation levels. The NO_2 concentrations are in the same order of magnitude but in contrast, they slowly start to rise in October. This can be explained by the beginning of colder weather with an uprise of motorized traffic and heating associated with combustion products. The same trend is valid for SO_2, the levels of which are one order of magnitude lower, below 5 ppb.

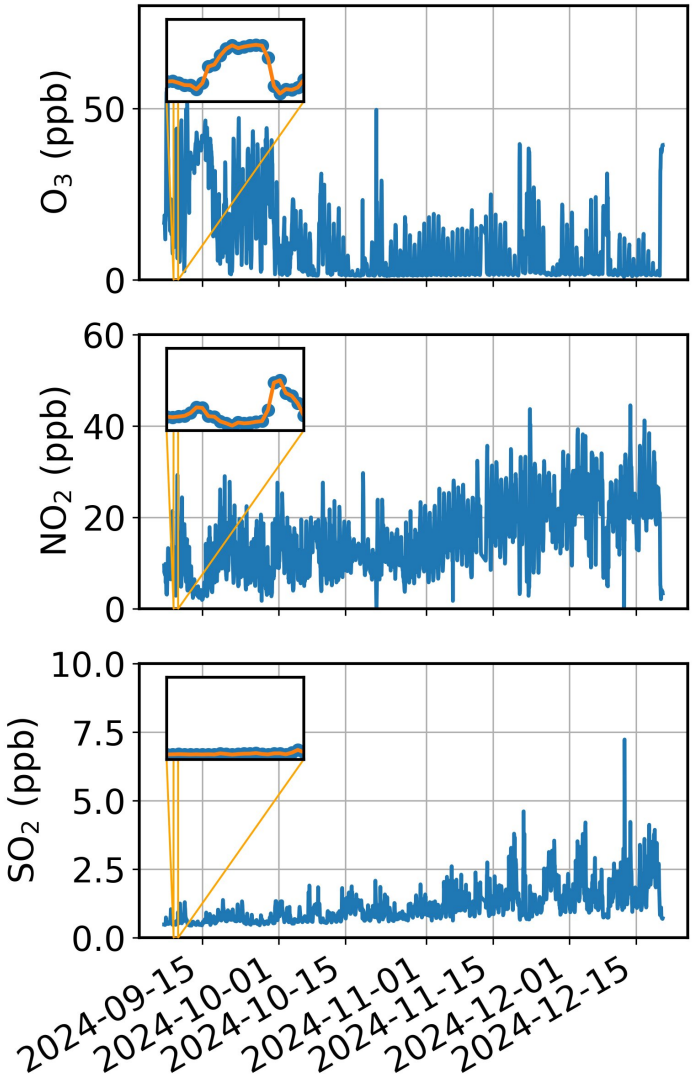


Figure 4.44: Time series of the pollutant concentrations in the Trento data set. The hourly reported values in blue are interpolated as highlighted by the orange curve in the insets.

The already approved data treatment (see Section 4.5) has been applied to the „Trento“ data set: A shuffled day-wise split into training, validation and test sets has been used to choose the most effective CNN. The resulting plots of predicted vs. true concentrations in the test set are shown in Figure 4.45. The average MRE in the test set is 25.0 %. The individual MRE of NO_2 and SO_2 is well below 20 %, while that of O_3 is almost twice as high as in the UBA data set. This result can be understood because the majority of the concentrations in the test set are below 20 ppb and generally, few concentrations above the LOD estimated in the UBA experiment of $\approx$ 14 ppb are encountered. In turn, the estimated LOD here is almost 40 ppb and many predictions are outside of the 25 % error margins.
The prediction of NO_2 shows perfect linearity with much fewer points outside the selected error margins with an estimated LOD around 9 ppb and RMSE around 4 ppb. The prediction of SO_2 is also remarkably accurate, given the absolutely lower concentrations. Similar to the findings in the GMS data set, high concentrations are underpredicted, though there are few of them in the overall sample population. The estimated LOD around 1 ppb and RMSE well below 1 ppb are quite low. SO_2's NRMSE is comparable to that of NO_2 around 9 % and is lower by a factor four than that of O_3.

Table 4.11: Summary of metrics for the prediction of pollutants in Trento.

Gas	LOD (ppb)	MAE (ppb)	RMSE (ppb)	MRE (%)	NRMSE (%)
NO_2	9.2	2.7	4.1	16.2	9.6
O_3	38.8	4.0	17.1	40.0	43.0
SO_2	1.1	0.3	0.4	18.7	8.9

The SHAP analysis visualized in Figure 4.46 attests an analogous working principle as in the ozone prediction in the UBA data set in Tübingen. Pixel 3 contributes most of the correctly interpreted information at the lower temperatures during the cool-down phase. These features, relevant for the ozone prediction, are also correlated to the prediction of the second oxidizing pollutant, NO_2. For the lowest concentration of the the latter, the last step of pixel 3 displays a high SHAP value, breaking the trend. These similarities of O_3 and NO_2 in terms of their sensing mechanism explain the errors most prevalent at low concentrations due to competition. A single step at medium temperature of the heat-up phase accounts for a clear prediction of O_3, significantly less pronounced for NO_2. While SO_2 also benefits from low temperature features, this is more dominant at the „dip“, joint by some of the high temperature plateaus. Additionally, pixel 2 and 4 chime in heavily, though there is not always a clear monotonous trend of a single feature visible.

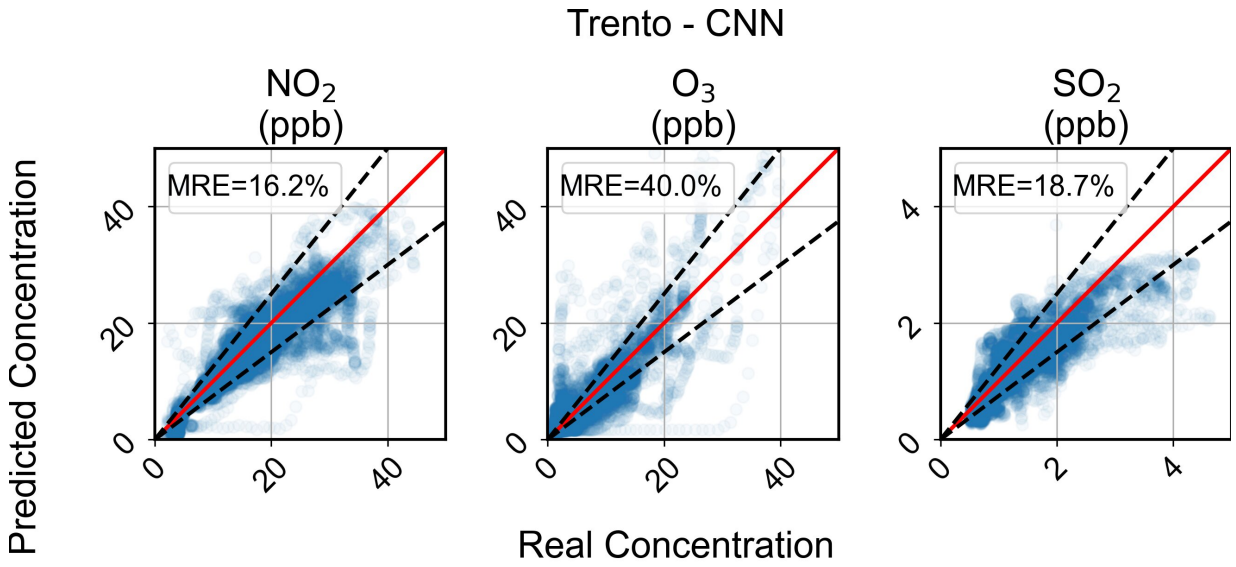


Figure 4.45: Pollutants in Trento predicted by a CNN.

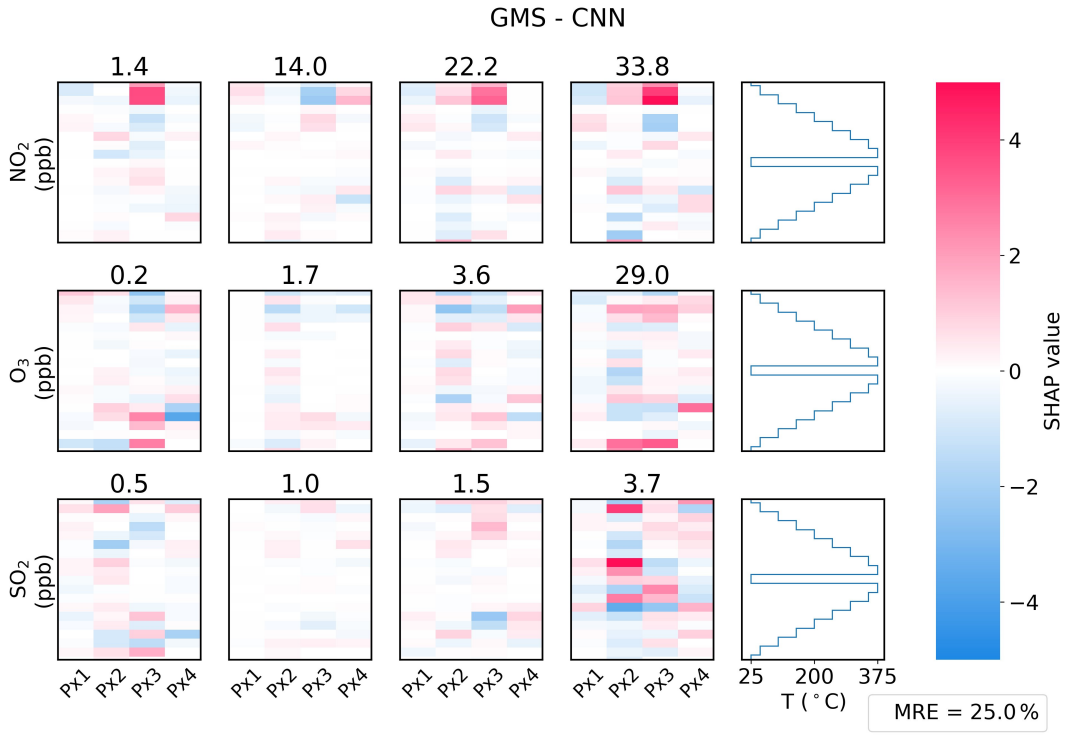


Figure 4.46: SHAP values for the pollutant prediction by a CNN with an SGP40 in Trento. The representative concentrations increase from left to right. The averaged and weighted SHAP values for each pixel (left to right) and temperature step in the modulation cycle (bottom to top) are visualized with a diverging blue-red color scale.

The excellent performance in the „Trento“ data set validates the general chemometric approach presented here. A single commercial gas sensor unit operated in temperature modulation can be utilized to predict pollutant gas concentrations in real outdoor air. The training of an advanced CNN model may take place on the foundation of publicly available reference data; e. g. by the EEA API. This concept was proven for two iterations of Sensirion's SGP sensors and in two urban locations in Europe with similar accuracy. Not only was the quantization of ozone possible during it's peak in summer, but it was also feasible in winter. Finally, the range of target gases has successfully been extended to NO_2 and SO_2 with even higher accuracy by developing a standalone device.

Future work could focus on the large scale production of these instruments and address related issues. Further miniaturization could open new application fields and facilitate automated production processes. The long term stability should be revisited and studies with a large batch of devices could focus on generating a general model to exclude or minimize sensor to sensor variation.

5 Summary and Outlook

The overall chemometric concept to detect and quantify pollutant gases with a single SMOX gas sensor array in temperature modulation has successfully been demonstrated in combination with advanced chemometric methods in a number of experiments and environments.
The first pilot study was the generation of the GMS data set, where the pollutant gases CO, NO_2, NO_2, and SO_2 have been dosed in randoms concentrations and mixtures matching those relevant for the definition of outdoor air quality guidelines by the WHO. These gas exposures under laboratory conditions have been enriched by a wide variation of RH, which is a known interferant present in virtually all applications in vastly varying absolute concentrations. An existing graphical software to automatically create these gaseous mixtures has been extended and finally been ported to Python to create a flexible software suite with improved source control and data base connection for the continuing gas sensor research. A commercial SGP30 gas sensor unit has been operated in temperature modulation between room temperature and 375 °C to yield a three minute long feature array of resistance responses. The chemometric models have been trained and optimized on data subsets, unrelated to a final test set, where the average MRE of prediction was around 10 %. While all models were able to predict the four pollutants and RH relatively well, the most complex of them, the CNN, produced the lowest errors consistent through different error metrics like MAE, MRE, and RMSE as well. The target gas with the highest accuracy was shown to be without doubt relative humidity with low single digit MREs. The CNN could especially boost the performance for the prediction of those gases that have higher relative errors in comparison. NO_2, O_3, and SO_2 all yield MREs above 10 % , but the predicted vs. true concentration plots reveal much lower relative errors than for the other model types. For these three target gases, the prediction of concentrations below 100 ppb produces relative errors often larger than 25 %. The detection of CO over the largest absolute concentration range give a remarkable MRE of 3.5 %. The accuracy of a CNN benefits from training to predict all gases simultaneously opposed to training one model to extract and interpret distinct features.
The same sensing approach, that has now been proven to work under well defined conditions, has been applied to real outdoor air and the generation of a second model data set. In this UBA set, the same type of SGP30 sensor has been exposed to air on the campus Morgenstelle in Tübingen and referenced to a public measurement station in the city. Due to the EOLs status of the SGP30, its successor the SGP40 has also been operated in the same fashion. The esp32ota project has been brought up to create hardware around the ESP32 microcontroller with flexible micropython software, bundling multiple drivers and creating easy to configure sensor nodes to jump start the operation and read out of (commercial) digital sensors. The concentration of ambient ozone in the summer season from June to August, 2022 has been as high as 60 ppb. Even in this lower concentration regime, a reasonable prediction of ozone by the proposed means is possible. Again, the CNN gives the best results with a MRE of 22.5 % roughly twice as high as that of O_3 in the GMS data set.

For both data sets, logarithmizing both the resistance and concentration data has been shown to be necessary for regression, which is in line with common practice and reasoned by several theoretical models. Additionally, shuffling the time series data before splitting the set into independent subsets has increased the final accuracy, pointing towards a certain sensor aging or drift. The SGP40 sensor, investigated in the UBA experiment in parallel, performed very comparably to its predecessor. The lower accuracy of ozone prediction in outdoor air could be owed to manifold reasons. First, the encountered concentrations are much lower and more often in the range of the LOD. Secondly, the matrix of real ambient air is much more complex than even the realistically modeled gas composition in the laboratory. It could also differ more than anticipated from the chosen reference station, which further resides at a different altitude. Moreover, the ambient temperature varies drastically – and with it the relative humidity – and has been above the lowest residence temperature of the temperature cycle at times. Still, these results provide sufficient motivation to further investigate and improve this type of analysis. For a long time experiment, a temperature mode with elevated start and end temperature of the measurement cycle has been chosen to be resilient against ambient temperature changes and hinted at by larger coefficients of the ridge regression. Over a course of one and a half years, a body of measurement data with roughly ten months of uninterrupted readings have been recorded this way. However, this temperature cycle with higher temperatures turned out to be unsuitable for the quantification of ozone with an SGP40. The CNN trained on shuffled data appeared to at least catch general trends, while that of unshuffled data showed systematic underprediction. This discrepancy suggests a certain aging of the sensor investigated, yet the comparison with a model trained on a shorter time frame – equivalent to the previous successful one – highlights the unfit operation mode.
To interpret the influence of the sensing materials' temperatures towards the prediction of target gases, single gas exposure sequences have been measured in synthetic air under laboratory conditions. Besides the already tested mode, some SGP40 sensors' operation temperature has been screened in a triangle wave. This type of analysis has not singled out individual temperatures or sensor pixels with a dominant response or sensor signal towards distinct gases. However, low temperatures in general resulted in the largest observable differences between pure air and the presence of target gases, especially CO and the oxidizing gases NO_2 and O_3.
A SHAP analysis has been carried out to further understand the individual sensing materials' and temperatures' contribution towards the (correct) prediction of target gas concentrations across the main data sets and different model types. For each data set and target gas, four representative concentrations covering the whole range have been evaluated. In general, the different models in one data set extract similar features, i. e. temperature steps of individual sensor pixels, for the prediction of the different gases. This finding validates the extraction of chemical information in the overall analytical scheme as the models do not simply fit noise. However, the extent and quality of feature extraction differs largely. In the GMS data set, the XGBoost regressor extracts few temperature steps and founds the regression on these. Most pronounced features are the low temperature steps of pixel 3 and 4 four the quantification of RH, CO, and O_3. NO_2 benefits from high temperature steps in the cool-down phase and SO_2 is hardest to detect accurately. Both, the ridge regression as well as the CNN model extract much more features, however, they also contain noise to some extent. This amount of diverging contributions towards the numeric prediction can be seen be SHAP values of opposite signs for the same samples. Still, the amount of extracted data is large and robust enough to increase the

performance. Especially the CNN can interpret a large number of features correctly.
In the UBA data set, a lot of attention is attributed towards low temperature steps for the correct detection of ozone. Not only does this differ from the behavior of O_3 in laboratory conditions, it also explains, why the truncated temperature cycle with elevated temperatures produced very noisy readings in the long term experiment. This insight highlights the potentials of the SHAP analysis for identifying relevant features, may it be sensing materials or temperatures, to improve performance in a given task or trim and simplify the measurement and evaluation scheme. As a proof of concept, a following correlation analysis has been conducted, evaluating which features produce SHAP values that correlate most positively with the actual concentrations. Using these results, the already acquired data has been trimmed artificially to train slimmer CNN models maintaining most of the original performance.
To battle some of the other shortcomings when dealing with real air samples, a novel modular sampling and measurement apparatus „eNose platform“has been developed jointly with *AO Action*, to provide a standalone device that can be operated offline and therefore at many places, including those closer to public reference stations. Further, a sensor module attachable to a Raspberry Pi, „SenPi“, has been developed to operate analog sensors in a similar fashion and broaden the selection of available sensors and sensing materials.
The eNose Platform with its SGP40 sensors has been in use in Tübingen for the second semester of 2024. An analysis in the same manner as the UBA experiment yields similar results and validates the device's operation principle once more. The „SenPi“ unit has been operated in the last quarter of 2024 and has thus experienced a time with low ambient ozone concentrations. As a results, no overwhelming prediction performance could be produced. Nevertheless, it was shown to measure resistances over five orders of magnitude and be applicable for operating analog sensors in temperature/voltage modulation.
A second eNose Platform has been deployed in the city center of Trento, Italy, directly next to a public reference station. Here, the measurement of NO_2 and SO_2 were available in addition to O_3, in the last quarter of 2024. In an analogous approach, the three gases could be predicted with an average MRE around 20 %. The MRE of about 40 % for ozone is comparable to that achieved with an equivalent unit in Tübingen. For NO_2 and SO_2 the MRE is well below 20 %. This experiment has validated not only the approach once more, but also the usability of the developed apparatus and its application towards the quantification of even more pollutant gases in a different location. To sum it up, the analytical process of determining the pollutant concentrations for outdoor air quality has been demonstrated with a temperature modulated SMOX sensor array and chemometric models, most importantly with CNNs. The concept has been proven in controlled laboratory conditions and successfully been ported to real outdoor air in multiple seasons and two locations in Europe. The developed software suite to control GMSs and sensor frame work around the ESP32 microcontroller will continue to aid gas sensor research in the author's work group. The eNose Platform finds further application in process and food control with multiple collaboration partners.

Outlook

Several of the concepts developed or shown in this work could be driven further. The SHAP analysis can be used to not only trim the feature space a posteriori to yield slimmer models, but to define and test completely new temperature cycles with the same goal and coupled with a higher sampling rate due to the shortened cycle. Moreover, dedicated modulation modes could be designed, matching individual target gases, improving the prediction in different applications. Ultimately, a new long term study based on a suitable operation mode could be conducted to identify and address sensor aging.
Another new field of research opens up by the development of modular sensor nodes, namely transferability. A larger batch of devices could be operated in an equivalent manner to inspect, how well models trained on one set are applicable to new units, or how much and what kind of recalibration is indeed necessary. Especially for the complex CNN models, transfer learning appears to be a feasible solution. Similarly, this exploration could be widened to a transfer not only from sensor to senor, but also from application to application.

6 Bibliography

[1] World Health Organization, *WHO global air quality guidelines: particulate matter (PM2.5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide*, Section: xxi, 273 p., World Health Organization, Geneva, **2021**.

[2] S. Osipov et al., *Communications Earth & Environment* **2022**, *3*, 203, https://www.nature.com/articles/s43247-022-00514-6.

[3] J. Fuhrer et al., *Ecology and Evolution* **2016**, *6*, 8785–8799, https://onlinelibrary.wiley.com/doi/10.1002/ece3.2568.

[4] O. R. Cooper et al., *Elementa: Science of the Anthropocene* **2014**, *2*, 29, https://online.ucpress.edu/elementa/article/doi/10.12952/journal.elementa.000029/112944/Global-distribution-and-trends-of-tropospheric.

[5] European Air Quality Index, %5Curl%7Bhttps://airindex.eea.europa.eu/AQI/index.html%7D (visited on 12/04/2024).

[6] I. O. for Standardization, ISO-13964:1998 - Ambient Air: Determination of Ozone: Ultraviolet Photometric Method, **1998**.

[7] N. Bârsan, U. Weimar in Proceedings of the 14th International Meeting on Chemical Sensors (IMCS 2012), Nürnberg/Nuremberg, Germany, **2012**, pp. 618–621.

[8] G. Heiland, *Zeitschrift für Physik* **1954**, *138*, 459–464, http://link.springer.com/10.1007/BF01340692.

[9] N. Taguchi, Gas-Detecting Device, en, tech. rep., Filed on July 14, 1970, and granted on December 28, 1971.

[10] G. Heiland, D. Kohl in *Chemical Sensor Technology*, Elsevier, **1988**, pp. 15–38, https://linkinghub.elsevier.com/retrieve/pii/B9780444989017500075.

[11] A. Staerz et al., *Sensors and Actuators B: Chemical* **2016**, *237*, 54–58, https://linkinghub.elsevier.com/retrieve/pii/S0925400516309236.

[12] D. Degler, U. Weimar, N. Bârsan, *ACS Sensors* **2019**, *4*, 2228–2249, https://pubs.acs.org/doi/10.1021/acssensors.9b00975.

[13] J. W. Gardner, P. N. Bartlett, *Sensors and Actuators B: Chemical* **1994**, *18*, 210–211, https://linkinghub.elsevier.com/retrieve/pii/0925400594870853.

[14] M. Ni, J. R. Stetter, W. J. Buttner, *Sensors and Actuators B: Chemical* **2008**, *130*, 889–899, https://linkinghub.elsevier.com/retrieve/pii/S0925400507009069.

[15] C. Bax et al., *Diagnostics* **2022**, *12*, 776, https://www.mdpi.com/2075-4418/12/4/776.

[16] K. Wu, M. Debliquy, C. Zhang, *Comprehensive Reviews in Food Science and Food Safety* **2023**, *22*, 913–945, https://ift.onlinelibrary.wiley.com/doi/10.1111/1541-4337.13095.

[17] J. Yan et al., *Sensors* **2015**, *15*, 27804–27831, https://www.mdpi.com/1424-8220/15/11/27804.

[18] A. P. Lee, B. J. Reedy, *Sensors and Actuators B: Chemical* **1999**, *60*, 35–42.

[19] A. Heilig et al., *Sensors and Actuators B: Chemical* **1997**, *43*, 45–51, https://linkinghub.elsevier.com/retrieve/pii/S0925400597000968.

[20] FIGARO Engineering Inc., TGS 3870-B00: For the detection of both methane and carbon monoxide, Data Sheet.

[21] S. Wold, *Chemometrics and Intelligent Laboratory Systems* **1995**, *30*, 109–115, https://linkinghub.elsevier.com/retrieve/pii/0169743995000429.

[22] P. Peng et al., *Sensors* **2018**, *18*, 157, http://www.mdpi.com/1424-8220/18/1/157.

[23] Y. Robin et al., *Atmosphere* **2021**, *12*, 1487, https://www.mdpi.com/2073-4433/12/11/1487.

[24] X. Zhai et al., *IEEE Access* **2016**, *4*, 8138–8146, http://ieeexplore.ieee.org/document/7605493/.

[25] J. Chu et al., *Sensors and Actuators B: Chemical* **2021**, *329*, 129090, https://linkinghub.elsevier.com/retrieve/pii/S0925400520314325.

[26] O. Djedidi et al., *Sensors and Actuators B: Chemical* **2021**, *339*, 129817, https://linkinghub.elsevier.com/retrieve/pii/S0925400521003865.

[27] J. Thorson, A. Collier-Oxandale, M. Hannigan, *Sensors* **2019**, *19*, 3723, https://www.mdpi.com/1424-8220/19/17/3723.

[28] X. Pan et al., *Sensors and Actuators B: Chemical* **2021**, *342*, 129982, https://linkinghub.elsevier.com/retrieve/pii/S0925400521005517.

[29] J. Yang, Z. Sun, Y. Chen, *Sensors* **2016**, *16*, 2069, http://www.mdpi.com/1424-8220/16/12/2069.

[30] Y. Ma et al., *ACS Sensors* **2024**, *9*, 6022–6031, https://pubs.acs.org/doi/10.1021/acssensors.4c01867.

[31] L. S. Shapley, *Notes on the N-Person Game — II: The Value of an N-Person Game*, RAND Corporation, Santa Monica, CA, **1951**, https://www.rand.org/pubs/research_memoranda/RM0670.html.

[32] S. M. Lundberg, S.-I. Lee in Proceedings of the 31st international conference on neural information processing systems, Curran Associates Inc., Red Hook, NY, USA, **2017**, pp. 4768–4777.

[33] V. Oliveira Santos et al., *Atmosphere* **2023**, *14*, 308, https://www.mdpi.com/2073-4433/14/2/308.

[34] C. Betancourt et al., *Geoscientific Model Development* **2022**, *15*, 4331–4354, https://gmd.copernicus.org/articles/15/4331/2022/.

[35] A. Hulanicki', S. Geab, F. Ingman, *Pure & Appl. Chem.* **1991**, *63*, 1247–1250.

[36] J. Vessman et al., Selectivity in Analytical Chemistry: (IUPAC Recommendations 2001), en, http://www.degruyter.com/view/IUPAC/iupac.73.0808.

[37] V. A. Gercher, D. F. Cox, *Surface Science* **1995**, *322*, 177–184, https://www.sciencedirect.com/science/article/pii/0039602895900284.

[38] S. Wicker et al., *The Journal of Physical Chemistry C* **2017**, *121*, 25064–25073, http://pubs.acs.org/doi/10.1021/acs.jpcc.7b06253.

[39] M. J. Madou, S. R. Morrison in *Chemical Sensing with Solid State Devices*, Elsevier, **1989**, pp. 479–516, https://linkinghub.elsevier.com/retrieve/pii/B978012464965150017X.

[40] N. Bârsan et al., *Journal of Electroceramics* **2010**, *25*, 11–19.

[41] N. Bârsan, *Sensors and Actuators B: Chemical* **1994**, *17*, 241–246, https://linkinghub.elsevier.com/retrieve/pii/092540059300873W.

[42] N. Bârsan, M. Hübner, U. Weimar, *Sensors and Actuators B: Chemical* **2011**, *157*, 510–517, https://linkinghub.elsevier.com/retrieve/pii/S0925400511004126.

[43] N. Bârsan, U. Weimar, *Journal of Electroceramics* **2001**, *7*, 143–167.

[44] S. Hahn et al., *Thin Solid Films* **2003**, *436*, 17–24, https://linkinghub.elsevier.com/retrieve/pii/S0040609003005200.

[45] *Gas sensors based on conducting metal oxides: basic understanding, technology and applications*, (Eds.: N. Bârsan, K. D. Schierbaum), Elsevier, Amsterdam, **2019**.

[46] N. Bârsan, U. Weimar, *Journal of Physics: Condensed Matter* **2003**, *15*, R813–R839, http://stacks.iop.org/0953-8984/15/i=20/a=201?key=crossref.cfe1b042be15201293045f0ecee08258.

[47] N. Yamazoe, K. Shimanoe, *Sensors and Actuators B: Chemical* **2008**, *128*, 566–573, https://linkinghub.elsevier.com/retrieve/pii/S0925400507005175.

[48] J. Fonollosa et al., *Procedia Engineering* **2015**, *120*, 261–264, https://linkinghub.elsevier.com/retrieve/pii/S187770581502264X.

[49] N. Bârsan et al., *Analytical Chemistry* **1999**, *71*, 2512–2517, https://pubs.acs.org/doi/10.1021/ac981246d.

[50] J. R. Stetter, W. R. Penrose, *Sensors Update* **2002**, *10*, 189–229.

[51] S. Morrison, *Sensors and Actuators* **1987**, *12*, 425–440, https://linkinghub.elsevier.com/retrieve/pii/0250687487800616.

[52] N. B. Tanvir et al., *Materials Today: Proceedings* **2015**, *2*, 4190–4195, https://linkinghub.elsevier.com/retrieve/pii/S2214785315007865.

[53] J. Burgués, S. Marco, *Sensors* **2018**, *18*, 339, http://www.mdpi.com/1424-8220/18/2/339.

[54] R. E. Cavicchi et al. in Conf. Proc. of Transducers '95 and Eurosensors IX, Stockholm, **1995**, pp. 823–826.

[55] H. Chai et al., *IEEE Sensors Journal* **2022**, *22*, 5470–5481, https://ieeexplore.ieee.org/document/9718073/.

[56] W. M. Sears, K. Colbow, F. Consadori, *Semiconductor Science and Technology* **1989**, *4*, 351–359, https://iopscience.iop.org/article/10.1088/0268-1242/4/5/004.

[57] X. Huang et al., *Sensors and Actuators B: Chemical* **2004**, *99*, 444–450, https://linkinghub.elsevier.com/retrieve/pii/S0925400503009018.

[58] A. Vergara et al., *Journal of Sensors* **2009**, *2009*, 1–10, http://www.hindawi.com/journals/js/2009/716316/.

[59] J. Zhang et al., *Journal of Materials Science: Materials in Electronics* **2016**, *27*, 4935–4942, http://link.springer.com/10.1007/s10854-016-4378-8.

[60] C. Schultealbert et al., *Sensors and Actuators B: Chemical* **2017**, *239*, 390–396, https://linkinghub.elsevier.com/retrieve/pii/S0925400516312278.

[61] Sensirion AG, SGP30 Version 1.0, Data Sheet, **2020**, https://sensirion.com/media/documents/984E0DD5/61644B8B/Sensirion_Gas_Sensors_Datasheet_SGP30.pdf (visited on 11/18/2024).

[62] Sensirion AG, SGP40 Version 1.2, Data Sheet, **2022**, https://sensirion.com/media/documents/296373BB/6203C5DF/Sensirion_Gas_Sensors_Datasheet_SGP40.pdf (visited on 11/18/2024).

[63] Daniel Rüffer, Felix Hoehne, Johannes Bühler, *Sensors* **2018**, *18*, 1052, http://www.mdpi.com/1424-8220/18/4/1052.

[64] B. S. Everitt, A. Skrondal, *The Cambridge Dictionary of Statistics*.

[65] B. Junker et al., *ACS Sensors* **2024**, *9*, 1584–1591, https://doi.org/10.1021/acssensors.4c00078.

[66] H. Robbins, S. Monro, *The Annals of Mathematical Statistics* **1951**, *22*, 400–407, http://projecteuclid.org/euclid.aoms/1177729586.

[67] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, **2017**, http://arxiv.org/abs/1412.6980.

[68] Y. Yao, L. Rosasco, A. Caponnetto, *Constructive Approximation* **2007**, *26*, 289–315, http://link.springer.com/10.1007/s00365-006-0663-2.

[69] Z. Hua et al., *Sensors and Actuators B: Chemical* **2018**, *255*, 1911–1919, https://www.sciencedirect.com/science/article/pii/S0925400517316192.

[70] Y. Xu, R. Goodacre, *Journal of Analysis and Testing* **2018**, *2*, 249–262, http://link.springer.com/10.1007/s41664-018-0068-2.

[71] D. A. Cicci, B. D. Tapley, *Celestial Mechanics* **1988**, *44*, 339–363, http://link.springer.com/10.1007/BF01234272.

[72] R. Tibshirani, *Journal of the Royal Statistical Society Series B: Statistical Methodology* **1996**, *58*, 267–288, https://academic.oup.com/jrsssb/article/58/1/267/7027929.

[73] F. Pedregosa et al., *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.

[74] T. Chen, C. Guestrin in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, **2016**, pp. 785–794, http://arxiv.org/abs/1603.02754.

[75] **2019**, DOI doi:10.1351/goldbook.L03540, https://doi.org/10.1351/goldbook.L03540.

[76] *DIN 32645: Nachweis-, erfassungs- und bestimmungsgrenze*, Beuth Verlag, Berlin, **2008**.

[77] F. Allegrini, A. C. Olivieri, *Analytical Chemistry* **2014**, *86*, 7858–7866, https://pubs.acs.org/doi/10.1021/ac501786u.

[78] Environmental Protection Agency, Washington, DC., Revised Assessment of Detection and Quantitation Approaches, en, tech. rep., **2004**.

[79] S. J. Prince, *Understanding deep learning*, MIT Press, **2023**, http://udlbook.com.

[80] H. Ismail Fawaz et al., *Data Mining and Knowledge Discovery* **2019**, *33*, 917–963, http://link.springer.com/10.1007/s10618-019-00619-1.

[81] Z. Wang, W. Yan, T. Oates, **2016**, http://arxiv.org/abs/1611.06455.

[82] A. Vaswani et al., **2017**, http://arxiv.org/abs/1706.03762.

[83] M. Krüger et al., *Natural Sciences* **2022**, *2*, e20220016, https://onlinelibrary.wiley.com/doi/abs/10.1002/ntls.20220016.

[84] Martín Abadi et al., TensorFlow: Large-scale machine learning on heterogeneous systems, **2015**, %5Curl%7Bhttps://www.tensorflow.org/%7D.

[85] H. Iqbal, HarisIqbal88/PlotNeuralNet v1.0.0, **2018**, https://zenodo.org/record/2526396 (visited on 10/30/2024).

[86] S. Ioffe, C. Szegedy, **2015**, http://arxiv.org/abs/1502.03167.

[87] S. Santurkar et al., **2019**, http://arxiv.org/abs/1805.11604.

[88] F. Allegrini, A. C. Olivieri, *Analytical Chemistry* **2016**, *88*, 7807–7812, https://pubs.acs.org/doi/10.1021/acs.analchem.6b01857.

[89] A. Pannone et al., *Nature* **2024**, DOI 10.1038/s41586-024-08003-w, https://www.nature.com/articles/s41586-024-08003-w.

[90] A. Dickow, G. Feiertag, *Procedia Engineering* **2014**, *87*, 1350–1353, https://linkinghub.elsevier.com/retrieve/pii/S1877705814028380.

[91] Y. Robin et al., *Atmosphere* **2022**, *13*, 1614, https://www.mdpi.com/2073-4433/13/10/1614.

[92] L. Völkl, MA thesis, University of Tuebingen, Tuebingen, **2024**.

[93] Gitlab, https://about.gitlab.com/ (visited on 01/17/2025).

[94] Preston-Werner, Semantic Versioning, **2013**, http://semver.org/ (visited on 01/17/2025).

[95] R. Dehuysser, micropython-ota-updater: A module to update your MicroPython code on the ESP8266/ESP32 via OTA, tech. rep., **2021**, https://github.com/rdehuyss/micropython-ota-updater.

[96] L. Pollak, Gilgamesh: A Python Package for Distributed Data, **2020**, https://pypi.org/project/gilgamesh/.

[97] L. Schynowski et al. in Proceedings of the 53rd International Conference on Environmental Systems, Louisville, Kentucky, **2024**, https://ttu-ir.tdl.org/bitstreams/92f84043-fafc-4377-afa8-19e3a941ee4d/download.

[98] S. Palzer, *Sensors* **2020**, *20*, 2745, https://www.mdpi.com/1424-8220/20/9/2745.

[99] A. Kobald, U. Weimar, N. Bârsan in 2022 IEEE International Symposium on Olfaction and Electronic Nose (ISOEN), IEEE, Aveiro, Portugal, **2022**, pp. 1–3, https://ieeexplore.ieee.org/document/9789650/.

[100] A. Kobald, U. Weimar, N. Bârsan in Eurosensors 2023, MDPI, **2024**, p. 178, https://www.mdpi.com/2504-3900/97/1/178.

[101] L. Schynowski, MA thesis, University of Tuebingen, Tübingen, **2022**.

[102] RPi HAT, https://github.com/raspberrypi/utils/tree/master/eeptools (visited on 04/29/2024).

[103] NXP, PCA9544A, en, Data Sheet, **2014**.

[104] MPS, MP8862 Rev 1.0, en, Data Sheet, **2019**, %5Curl%7Bhttps://www.monolithicpower.com/en/mp8862.html%7D.

[105] TI, ADS111x, Data Sheet, **2018**, %5Curl%7Bhttps://www.ti.com/lit/ds/symlink/ads1115.pdf?ts=1714365940842%7D.

[106] TI, LOG114, Data Sheet, **2007**, %5Curl%7Bhttps://www.ti.com/lit/ds/symlink/log114.pdf?ts=1714497935232%7D.

[107] JEDEC, Transistor Outlines Archive, tech. rep., JEDEC, **2001**, http://www.jedec.org/sites/default/files/docs/archive/to/to-012.pdf.

[108] D. Degler et al., *The Journal of Physical Chemistry C* **2015**, *119*, 11792–11799, http://pubs.acs.org/doi/10.1021/acs.jpcc.5b04082.

[109] J. Burgués et al., *iScience* **2021**, *24*, 103371, https://linkinghub.elsevier.com/retrieve/pii/S2589004221013420.

[110] C. Jaeschke et al., *Proceedings* **2018**, *2*, 993, http://www.mdpi.com/2504-3900/2/13/993.

[111] S. Kapoor, A. Narayanan, *Patterns* **2023**, *4*, 100804, https://linkinghub.elsevier.com/retrieve/pii/S2666389923001599.

[112] G. E. P. Box, G. C. Reinsel, G. M. Jenkins, *Time series analysis : forecasting and control*, Prentice-Hall, Englewood Cliffs, NJ, **1994**.

[113] A. Kobald.

[114] A. Alharbi et al., *Sensors* **2021**, *21*, 8462, https://www.mdpi.com/1424-8220/21/24/8462.

[115] B. Mikrotechnik, mp6 series Version 1.2, Data Sheet, **2023**.

[116] European Environment Agency's home page, en, %5Curl%7Bhttps://www.eea.europa.eu/en%7D.

[117] A. Gaiardo et al. in Eurosensors 2023, MDPI, **2024**, p. 36, https://www.mdpi.com/2504-3900/97/1/36.

[118] M. Magoni et al. in Eurosensors 2023, MDPI, **2024**, p. 33, https://www.mdpi.com/2504-3900/97/1/33.

[119] M. of Environmental Protection of the People's Republic of China, Ambient air quality standards, **2012**.

[120] S. Wang, J. Hao, *Journal of Environmental Sciences* **2012**, *24*, 2–13, https://linkinghub.elsevier.com/retrieve/pii/S1001074211607249.

[121] K. A. Mar et al., *Environmental Science & Policy* **2022**, *134*, 127–136, https://linkinghub.elsevier.com/retrieve/pii/S1462901122001204.

[122] Statista, Atmospheric Concentration of Methane (CH_4) Globally from 1990 to 2024, **2024**, https://www.statista.com/statistics/1314344/atmospheric-concentration-of-ch4-historic-monthly/.

[123] C. Schultealbert et al., *Atmosphere* **2021**, *12*, 366, https://www.mdpi.com/2073-4433/12/3/366.

[124] R. Derwent, P. Kay, *Environmental Pollution* **1988**, *55*, 191–219, https://linkinghub.elsevier.com/retrieve/pii/0269749188901522.

[125] J. C. Farman, B. G. Gardiner, J. D. Shanklin, *Nature* **1985**, *315*, 207–210, https://doi.org/10.1038/315207a0.

[126] M. Mueller, J. Meyer, C. Hueglin, Design of an ozone and nitrogen dioxide sensor unit and its long-term operation within a sensor network in the city of Zurich, en, preprint, Gases/In Situ Measurement/Instruments and Platforms, **2017**, https://amt.copernicus.org/preprints/amt-2017-22/amt-2017-22.pdf.

[127] L. Spinelle et al., *Sensors and Actuators B: Chemical* **2017**, *238*, 706–715, https://linkinghub.elsevier.com/retrieve/pii/S092540051631070X.

[128] J. Fonollosa, *ECS Meeting Abstracts* **2020**, *MA2020-01*, 1845, https://doi.org/10.1149/MA2020-01261845mtgabs.

[129] The Hammond Times, *The Hammond Times* **1957**, Hammond, Indiana, USA, 65.

[130] T. S. Breusch, A. R. Pagan, *Econometrica* **1979**, *47*, 1287, https://www.jstor.org/stable/1911963?origin=crossref.

[131] R. D. Cook, S. Weisberg, *Biometrika* **1983**, *70*, 1–10, https://doi.org/10.1093/biomet/70.1.1.

[132] S. Seabold, J. Perktold in 9th python in science conference, **2010**.

[133] J. Adebayo et al., Sanity Checks for Saliency Maps, en, **2020**, http://arxiv.org/abs/1810.03292.

[134] N. Yamazoe, N. Miura in *Chemical sensor technology*, (Ed.: S. YAMAUCHI), Elsevier, Amsterdam, **1992**, pp. 19–42, https://www.sciencedirect.com/science/article/pii/B9780444986801500073.

[135] T. Hitzler, Bachelor Thesis, University of Tuebingen, Tübingen, **2022**.

[136] S. M. Lundberg et al., *Nature Machine Intelligence* **2020**, *2*, 56–67, https://www.nature.com/articles/s42256-019-0138-9.

[137] FIGARO Engineering Inc., TGS2600, Data Sheet.

[138] H. Aono et al., *Sensors and Actuators B: Chemical* **2003**, *94*, 132–139, https://linkinghub.elsevier.com/retrieve/pii/S0925400503003289.

[139] J. N. Miller, J. C. Miller, *Statistics and chemometrics for analytical chemistry*, 6th ed, Prentice Hall/-Pearson, Harlow, **2010**.

Glossary

R^2 Coefficient of Determination

args Arguments – A list of python arguments typically unpacked as *args

docstring Documentation String

eNose Platform eNose Platform (AO-Action)

epoch Absolute date and time measured in seconds passed since January 1st, 1970 00:00:00 UT

ESP32 ESP32 microcontroller (Espressif)

esp32ota Micropython framework for the ESP32 to control and read digital (gas) sensors

GND Ground, electrical reference point in a circuit

kwargs Keyword arguments – A dictionary of python keyword arguments typically unpacked as **kwargs

PyGUI Python GUI for controlling experimental setups

PyQt Python bindings for the GUI toolkit Qt

Acronyms

Adam Adaptive Momentum Estimation

ADC Analog to Digital Converter

ANN Artificial Neural Network

API Application Programming Interface

ASIC Application-Specific Integrated Circuit

CART Classification And Regression Tree

CI Confidence Interval

CNN Convolutional Neural Network

CRC Cyclic Redundancy Check

csv Comma Separated Values

CV Computer Vision

DNN Deep Neural Network

DOE Design Of Experiment

edf European Data Format

EEA European Environment Agency

EEPROM Electrically Erasable Read Only Memory

EOL End Of Life

EPA (United States) Environmental Protection Agency

FBK Fondazione Bruno Kessler, Italy

GMS Gas Mixing System

GPIO General Purpose Input/Output

GUI Graphical User Interface

HAT Hardware Attached on Top

I2C Inter-Integrated Circuit

IAQ Indoor Air Quality

IDC Insulation-Displacement Contact

IPC Institute of Physical Chemistry (of the University of Tübingen)

ISS International Space Station

IUPAC International Union of Pure and Applied Chemistry

json JavaScript Object Notation

LASSO Least Absolute Shrinkage and Selection Operator

LOD Limit of Detection

MAE Mean Absolute Error

MEMS Micro-Electro-Mechanical Systems

MFC Mass Flow Controller

ML Machine Learning

MLR Multiple Linear Regression

MRE Mean Relative Error

MSE Mean Square Error

NDIR NonDispersive InfraRed

NN Neural Network

NRMSE Normalized Root Mean Square Error

NTC Negative Thermal Coefficient

OAQ Outdoor Air Quality

OLS Ordinary Least Squares

OpAmp Operational Amplifier

ota over-the-air

PCA Principal Componenet Analysis

PCB Printed Circuit Board

PEEK Polyether ether ketone

PI Prediction Interval

PM Particulate Matter

PTC Positive Thermal Coefficient

PTFE Polytetrafluoroethylene

RH Relative Humidity

RMSE Root Mean Square Error

RPi Raspberry Pi

RTC Real Time Clock

SCL Serial CLock

SDA Serial DAta

SenPi Sensor Raspberry Pi

SFO Samarium Iron Oxide

SGD Stochastic Gradient Descent

SHAP SHapley Additive exPlanations

SMOX Semiconducting Metal Oxide

SPI Serial Peripheral Interface

TO-12 Transistor Outline 12

UART Universal Asynchronous Receiver / Transmitter

UBA Umweltbundesamt – a German environment protection agency

WDT Watch Dog Timer

WHO World Health Organization

XGBoost EXtreme Gradient Boosting

Publications

Articles in Peer Reviewed Journals as First Author

[1] A. Kobald, U. Weimar, N. Bârsan, Towards Understanding Temperature Modulated SMOX Gas Sensor Arrays for Outdoor Air Quality Applications, **2025**, `https://www.sciencedirect.com/science/article/pii/S0925400525006549`.

Articles in Peer Reviewed Journals as Co-Author

[1] A. Staerz, A. **Kobald**, T. Russ, U. Weimar, A. Hémeryck, N. Bârsan, "Thermal Water Splitting on the WO_3 Surface: Experimental Proof", *ACS Applied Electronic Materials* **2020**, *2*, 3254–3262.

[2] Y. Liu, U. Geyik, A. **Kobald**, A. Yang, X. Wang, U. Weimar, M. Rong, N. Bârsan, "Overheat Diagnosis of Power Cable Based On Gas Sensors: Device/Material Exploration", *Sensors and Actuators B: Chemical* **2021**, 130837.

[3] S. Braun, A. **Kobald**, A. Oprea, I. Boehme, P. Bonanati, U. Weimar, N. Bârsan, "Monitoring hand hygiene with commercial gas sensors: A pattern recognition approach", *Sensors and Actuators B: Chemical* **2022**, *352*, 131027.

[4] B. Junker, A. **Kobald**, C. Ewald, P. Janoschek, M. Schalk, U. Weimar, L. Mädler, N. Bârsan, "Multivariate Analysis of Light-Activated SMOX Gas Sensors", *ACS Sensors* **2024**, *9*, 1584–1591.

[5] L. Schynowski, M. Schalk, A. **Kobald**, F. Meyer, C. Eigenbrod, L. Mädler, U. Weimar, N. Bârsan, "Development of a pre-ignition fire detection sensor system", **2024**, ICES-2024-98.

Presentations at Conferences

[1] A. Kobald, U. Weimar, N. Bârsan, "Regression Model for the Prediction of Pollutant Gas Concentrations with Temperature Modulated Gas Sensors", **2022**, 1–3.

[2] A. Kobald, U. Weimar, N. Bârsan, "Prediction of Atmospheric Ozone Concentrations with a Temperature-Modulated Gas Sensor Array", **2024**, 178.

Appendix

PyGUI

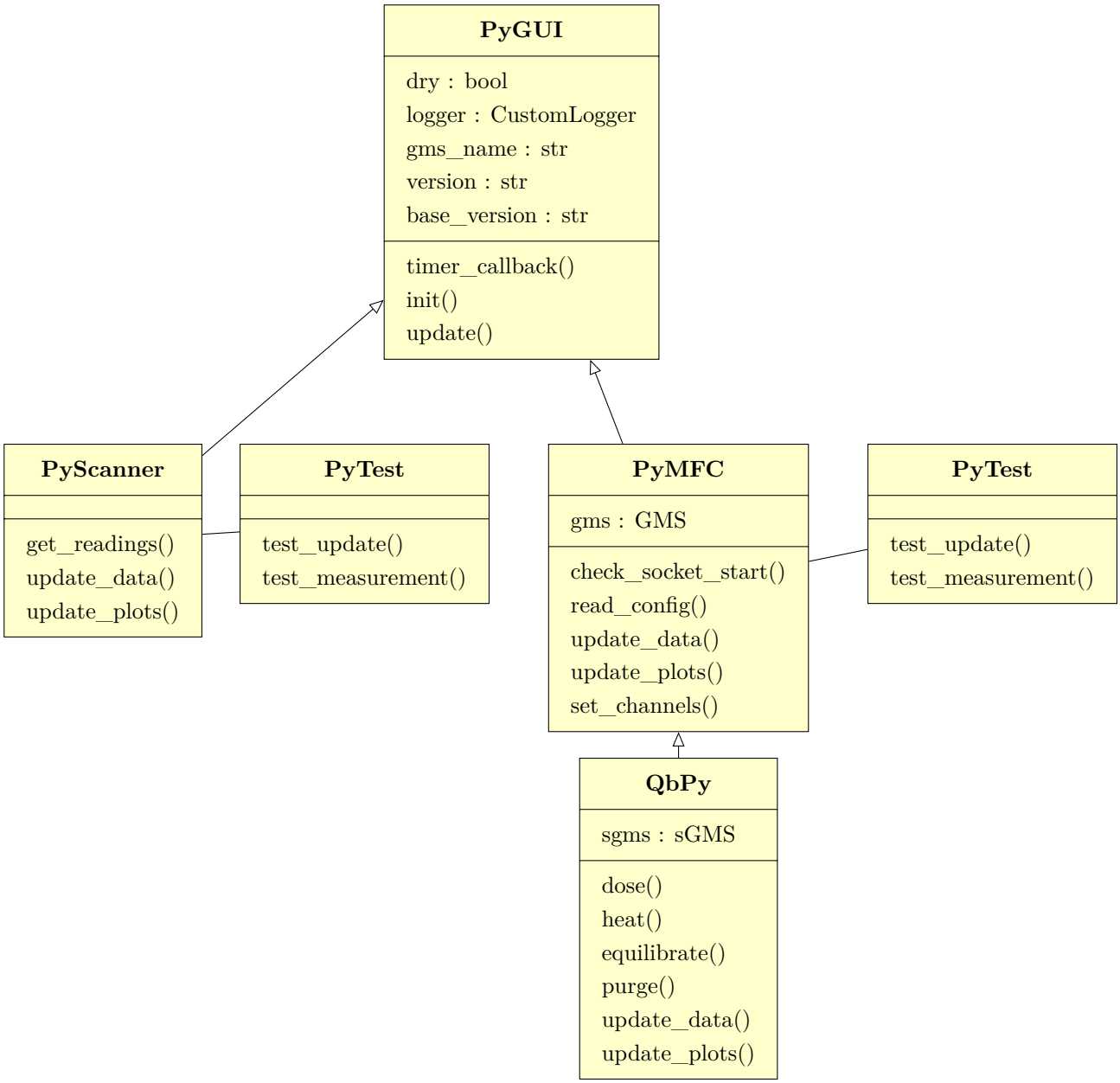


Figure 6.1: Family tree of the different GUIs with selected attributes and simplified methods.

eNose Platform

Table 6.1: Low level functions implemented on the eNose Platform.

set_pump(pump_id=0, value=False)

Set a micropump to a given Value.

pump_id: *int*, optional
The id of the pump to control, either 0 or 1. Default 0.

value: *bool*, optional
The state to set the pump to. Default False (Off).

led(color='off')

Set the status LED to one of the possible colors.

color: *str*, optional
The wanted LED color. One in ['red', 'green', 'yellow', 'off'].
Default 'off'.

Individual Models per Gas – GMS

Table 6.2: Optimized hyperparameters of the CNN models trained to predict a single gas in random mixtures in the GMS data.

Hyperparameter	All Gases	CO	NO_2	O_3	SO_2	RH
Kernel Size	6	6	4	5	6	4
# of Filters in 1st Conv2D Layer	32	32	128	32	64	64
# of Neurons in Dense Layer	128	64	128	128	64	64
L_2 (Kernel, Bias, Activity)	$(5 \times 10^{-3}, 5 \times 10^{-3}, 5 \times 10^{-3})$	$(5 \times 10^{-3}, 1 \times 10^{-2}, 5 \times 10^{-3})$	$(1 \times 10^{-2}, 1 \times 10^{-2}, 1 \times 10^{-2})$	$(5 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-1})$	$(5 \times 10^{-3}, 5 \times 10^{-3}, 5 \times 10^{-3})$	$(5 \times 10^{-3}, 1 \times 10^{-2}, 5 \times 10^{-3})$
Dropout	5×10^{-3}	0	0	5×10^{-3}	0	0
Batch Normalization	True	True	True	True	True	True

Ozone – SGP40

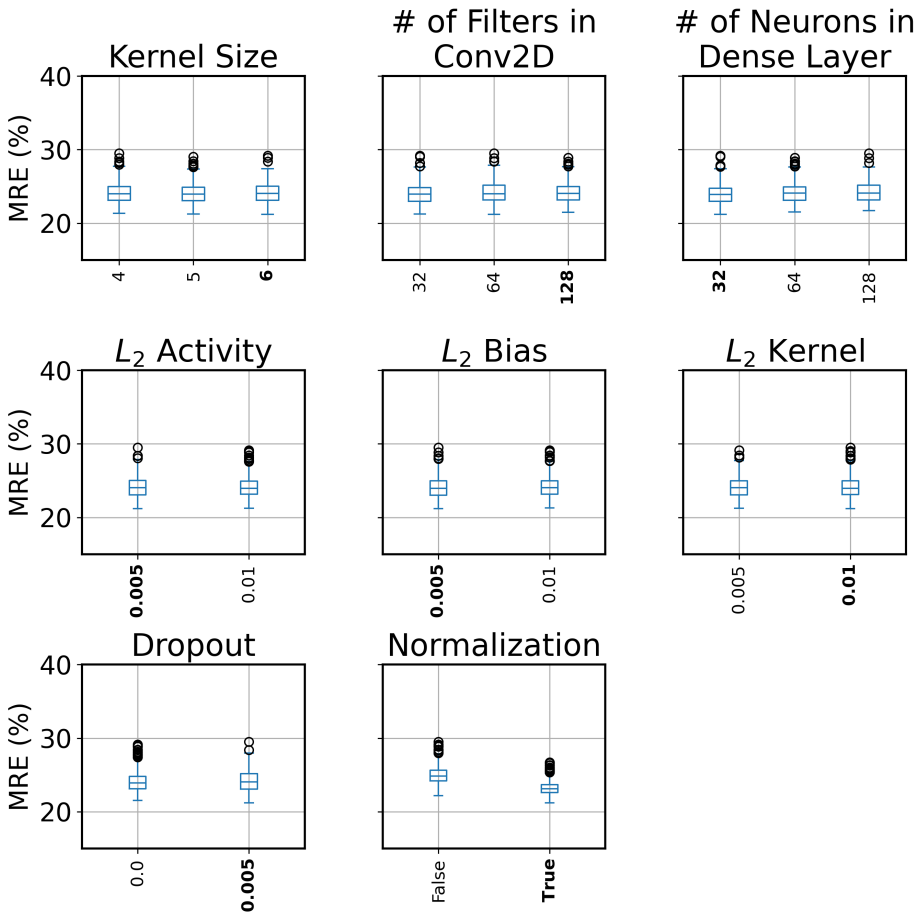


Figure 6.2: Univariate influence of hyperparameters on the MRE of predicted concentrations for the validation UBA data with an SGP40. Selected choices are highlighted in bold font.

Table 6.3: Selection of hyperparameters of the investigated CNN models in the UBA data set for SGP30 and SGP40 sensors.

Hyperparameter	Values	Selected for SGP30	Selected for SGP40
Kernel Size	4, 5, 6	6	5
# of Filters in 1st Conv2D Layer	32, 64, 128	32	128
# of Neurons in Dense Layer	32, 64, 128	128	32
L_2 (Kernel, Bias, Activity)	1×10^{-2}, 5×10^{-3}	(5×10^{-3}, 5×10^{-3}, 5×10^{-3})	(1×10^{-2}, 5×10^{-3}, 5×10^{-3})
Dropout	0, 1×10^{-2}, 5×10^{-3}	5×10^{-3}	5×10^{-3}
Batch Normalization	True, False	True	True