Sensor Validation for Indoor Air Quality using Machine Learning

Vagner Seibert, Ricardo Matsumura Araújo[1], Richard McElligott

[1]Universidade Federal de Pelotas (UFPEL) – Pelotas, RS – Brazil

vagnerseibert@gmail.com, ricardo@inf.ufpel.edu.br, richard@futuredecisions.net

Abstract. *To guarantee a high indoor air quality is an increasingly important task. Sensors measure pollutants in the air and allow for monitoring and controlling air quality. However, all sensors are susceptible to failures, either permanent or transitory, that can yield incorrect readings. Automatically detecting such faulty readings is therefore crucial to guarantee sensors' reliability. In this paper we evaluate three Machine Learning algorithms applied to the task of classifying a single reading from a sensor as faulty or not, comparing them to standard statistical approaches. We show that all tested machine learning methods – Multi-layer Perceptron, K-Nearest Neighbor and Random Forest – outperform their statistical counterparts, both by allowing better separation boundaries and by allowing for the use of contextual information. We further show that this result does not depend on the amount of data, but ML methods are able to continue to improve as more data is made available.*

1. Introduction

Air pollution is considered a major environmental health risk in modern days. Excessive levels of harmful gases and particulate matter present in the air are responsible for about seven million deaths globally per year [World Health Organization 2020].

Control of a building Heating Ventilation and Air Conditioning (HVAC) systems enable the intake reduction of air pollutants to help manage the internal air quality that the building occupants are exposed. To enable such a task, indoor air pollution sensors are required. Laboratory grade air quality sensors are not practical due to both size and cost (e.g. [Mead et al. 2013]), where often several units would be required per pollutant measured at each intake with additional sensors required within the buildings space.

Building managers often make use of more affordable sensor packages for managing Temperature, Humidity and CO_2, however few buildings control for air pollutants (NO_2, O_3, NO, SO_2, PM_1, $PM_{2.5}$, PM_{10}). Electrochemical sensors are typically the most practical for gas measurements (cost, size, accuracy) and are usually combined with laser based measurements for particulate matter. Those are often calibrated with respect to laboratory grade equipment and are expected to provide equivalence with respect to exposure guidelines and legal air quality regulations [Mead et al. 2013].

Electrochemical sensors are nevertheless prone to failures and drift as they degrade over their expected lifetime. Drift and degradation can be corrected for, although non-catastrophic, transient failure can result in erroneous data. Such data should not be utilised within the building control system as it may lead to increased pollution exposure to building occupants. Erroneous air quality sensor data must also be excluded when data is utilised to meet obligations such WHO air quality guidelines or local air quality legislation [Muller et al. 2018].

To assess their correct operation, sensor validation was conducted. One of its aspects is detecting when a faulty measurement occurred. As such, it can be seen as a classification task. While some readings can be easily classified as faulty (e.g. a negative reading for some pollutant), others may fall within plausible value ranges. Several approaches try to tackle this issue. A common used method is to use statistical analysis. For example, one can calculate the mean and standard deviation of readings over a period of time and then new readings that fall outside two standard deviations are considered errors. These approaches typically are univariate, working directly on the sensor reading and not taking into account other contextual variables, such as temperature, which can influence a reading.

In this paper, we evaluate the use of Machine Learning (ML) techniques to train models on the task of classifying Indoor Air Quality sensor readings as correct or faulty. Our expectation is that such models can have two advantages over statistical approaches: (i) they are able to model more complex decision boundaries between correct and faulty readings and (ii) they can benefit from the use of contextual information.

In order to do so, we collected a total of 2,151,483 data points from an indoor air quality sensor network located in a commercial building situated in London, UK. An expert labeled a large sample of data points and we evaluated three ML algorithms (Multilayer Perceptron, K Nearest-Neighbors and Random Forests) trained over this data, comparing to traditional techniques.

We provide three main contributions in this paper:

(i) We show that ML models can outperform standard statistical approaches, including the one being currently used with the sensors considered here;
(ii) We show that ML can benefit from contextual information in addition to the sensors' main readings, and this is responsible for part of the observed improvements;
(iii) We show that ML models do not require more data than the statistical approaches to attain similar results, but are able to provide better results given additional data.

The rest of the paper is organized as follows. Section 2 describes some close related works and differentiate ours from them. Section 3 details the objectives, goals, data and methodology used for the experiments. Section 4 shows the obtained results from the proposed methods, including a comparison of the tested algorithms, and also an ablation study. Finally, Section 5 offers our conclusions and provide lines for future work.

2. Related Works

Sensor validation has been a topic of research for a long time, and play a role in increasing the reliability of industrial, environmental and chemical process monitoring [Upadhyaya and Eryurek 1992]. Also, flight control systems and smart buildings require accurate sensor monitoring to assure their correct operation [Napolitano et al. 1998].

There are two major approaches to sensor validation [Upadhyaya and Eryurek 1992]: statistical methods and knowledge based systems. Statistical methods are usually easier to implement and require less computational power, being often implemented within the sensor's hardware for self-validation. These include simple analysis of statistical descriptors (as will be explained later in this paper), but also use of Principal Component Analysis and variations when the right type of data is available [Ibarguengoytia et al. 2001, Kerschen et al. 2004, Friswell and Inman 1999]. Our work makes use of common statistical methods as the baseline for comparison.

Knowledge based systems comprehend methods that make use of heuristic reasoning, often requiring historical data, from which they generate the necessary models for predictions [Henry and Clarke 1993]. Machine Learning (ML) methods are included in this category. A Machine Learning method fit a complex model to historical data aiming at generalizing to unseen data. Common models include neural networks, support vector machines and decision trees. For example, in [Upadhyaya and Eryurek 1992, Mattern et al. 1998, Napolitano et al. 1998], neural networks are used for monitoring sensors in power plants. K-Nearest Neighbors, another ML model, is also used for fault detection in sensors (e.g. [Yang et al. 2016]). Our work also makes use of common ML models, but applied to the specific setting of Indoor Air Quality sensors.

More recently, several works propose the usage of Recurrent Neural Networks or Convolutional Neural Networks to analyse a time-series data to detect faults [Loy-Benitez et al. 2020, Gupta et al. 2020, Eren 2017], with promising results. These models are however quite complex, often requiring dedicated hardware for training and inference. Unlike these works, ours focus on point-wise detection – perform a classification from a single point of data, instead of a full stream.

3. Objectives, Data and Methods

Our main objective is to evaluate the efficacy of ML models applied to the task of identifying faulty readings from air quality sensors. We establish as our goals:

(i) To compare performance of different ML models to commonly used techniques based on simple statistics,
(ii) To evaluate the benefit of using contextual data (temperature) in addition to the main reading, and
(iii) To evaluate how much data is required to achieve adequate results.

3.1. Data

Data was provided by a company responsible for managing information from several sensors in business buildings in the United Kingdom. The provided sample was extracted from seven *pods* – a pod is a physical unit enclosing a set of sensors measuring one or more chemical components from the air as well as weather measurements. The data contains information from four gas concentrations: NO_2, O_3, CO and SO_2 and was collected every 10 minutes between October and November, 2017.

A data point is composed of a timestamp, a real-valued reading for a specific gas, a real-valued temperature at the moment of the reading and a categorical variable indicating the type of gas. A total of 2,151,483 data points were provided. Of those, 56,605 were provided with labels, 35,034 being faulty (38%), and 21,571 being normal (62%). These labels were attributed by a human expert and not automatically attributed.

Table 1. Example of labeled data points. Reading is the measurement value for a specific gas, temperature is the temperature (Celsius) at the time of the reading, type indicates the gas type and label indicates whether the reading was considered faulty (1) or normal (0).

Timestamp	Reading	Temperature	Gas	Label
2017-09-08 10:20:00	21.80	17.0	O3	0
2017-08-27 02:30:00	-1602.25	20.1	CO	1
2017-11-02 01:40:00	34282.17	12.5	O3	1
2017-08-27 00:00:00	466.63	21.5	NO2	0
2017-11-02 13:20:00	-0.043	13.5	O3	0

Table 2. Number of labeled data points for each type of gas.

Gas	Total (%)	Normal	Faulty	Normal (%)	Faulty (%)
CO	23.3%	11386	1831	86.1%	13.9%
NO_2	18.4%	4940	5496	47.3%	52.7%
O_3	40.9%	13803	9350	59.6%	40.4%
SO_2	17.3%	4905	4894	50.1%	49.9%
Total	100%	35034	21571	61.9%	38.1%

Fig. 1 and Table 1 show examples of labeled readings. Table 2 describes the data in detail by gas type and per label. It is possible to observe that except for CO, labels are quite balanced between classes.

3.2. Models

We evaluate a total of five classifiers - two statistical approaches and three Machine Learning approaches. Statistical approaches are often used to detect faulty readings and we include the one currently being used in production and a variation. The ML approaches were chosen due to being common in the literature while having simple implementations, often a requirement for deployment.

Mean: this method calculates the mean μ and standard deviation σ from the training set, using only readings labeled as normal. New data i is standardized by the Z-Score $(x_i - \mu)/\sigma$ and if its magnitude is above 2σ it is classified as faulty. Only sensors readings are used. This is the method currently being used for fault detection for the sensors considered in this study.

Median: this method calculates the median μ and interquartile range IQR of the training set, also only using readings labeled as normal. New data that falls below $Q1 - 2IQR$ or above $Q3 + 2IQR$ is classified as faulty. Only sensors readings are used. This was included due to being a common technique for outlier detection.

Multi-Layer Perceptron (MLP) [Kubat 1999]: a feedforward hierarchical artificial neural network composed of H hidden layers each with n_h neurons. The input layer is composed of three neurons, receiving as input the type of gas, the sensor reading (x) and the temperature (T). Output is a single logistic unit indicating whether the reading is faulty or not.

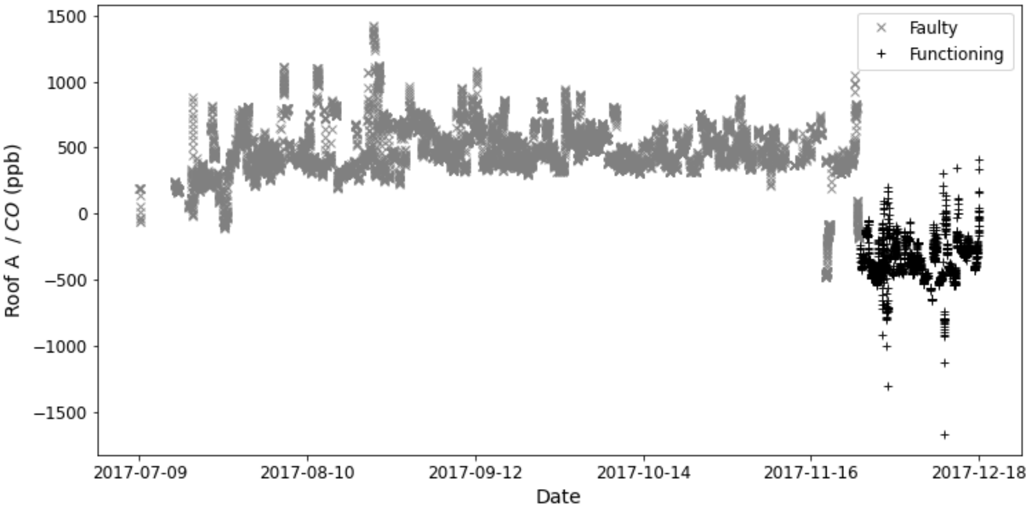

Figure 1. Sample of 10,129 data points retrieved from the pod Roof A, for the Carbon Monoxide sensor, from October of 2017 until November of 2017.

K-Nearest Neighbor (KNN) [Goldberger et al. 2005]: a lazy learning approach where the training data is stored and new data is attributed the class of the majority of its K nearest examples. A distance metric is used to determine the neighborhood.

Random Forest [Ho 1995]: an ensemble method that outputs the majority voting among D decision trees. Each decision tree is trained on a random sample taken from the training set in order to reduce overfitting.

3.3. Experiments

In order to evaluate and compare the chosen methods, we use the area under the Receiver Operating Characteristic (ROC) curve - AUC - as the metric of choice. This metric was chosen because it provides a simple way to rank the models' performance regardless of the class distribution, being invariant to prior class probabilities [Bradley 1997].

We use stratified k-fold cross validation [Blum et al. 1999] with $k = 20$. This procedure partitions the data into k equally sized folds, and then train k independent models using data from $k - 1$ folds, testing on the remaining fold. We report on the mean and standard deviation of the measured AUC on the k folds. Folds are stratified by label (faulty or not) so that each partition matches the observed label distribution across the whole dataset.

For the statistical approaches, separate models were trained for each gas type and we present the average results. This is necessary since readings' values vary considerably across gases and these techniques do not allow for the type of gas to be specified as an input. For the ML approaches, two experiments were conducted: (i) we trained one model per gas type and (ii) we trained a single model for all gas types, adding an extra input to the model specifying the gas type. The latter approach was conducted in order to test the ability of models to multi-task. Having a single model consume less resources, an important feature especially for embedded systems.

Table 3. Comparison of classification AUC of the presented algorithms using k-fold cross validation with $k = 20$.

Classifier	O_3	NO_2	SO_2	CO	Avg AUC	σ	Multitask AUC
Mean	0.92	0.87	0.81	0.95	0.89	0.062	N/A
Median	0.94	0.91	0.91	0.96	0.93	0.028	N/A
MLP	0.97	0.94	0.97	0.97	0.96	0.013	0.93
KNN	0.97	0.96	0.97	0.97	0.97	**0.007**	0.96
RF	**0.97**	**0.96**	**0.97**	**0.98**	**0.97**	0.008	**0.97**

In another experiment, we trained ML models both with and without the inclusion of Temperature in the input. Again, the statistical approaches do not allow for extra inputs in a straightforward way and we aim at testing if the extra context added leads to better performance.

As a last experiment, we varied the amount of data available to the classifiers. We randomly sample the dataset for each given size, which varied from 10 to 40,000 data points. For each size, 20 independent executions were conducted in order to compute the average AUC.

All hyper-parameters for ML models were set to the default values in the scikit-learn Python library (version 0.23.1) [Pedregosa et al. 2011]. The MLP uses a single hidden layer with 100 ReLu units, ADAM optimizer, initial learning rate of 0.001 and mini-batches of size 200 and 200 epochs. KNN uses Euclidean distance, $K = 5$ and uniform weight. Random Forest uses Gini coefficient for splits and 100 trees.

4. Results

Table 3 shows the results of training one model per gas type, with ML models receiving not only gas concentrations but also the temperature at the time of reading. We can observe that all ML algorithms outperformed the traditional methods by a reasonable margin, regardless of what gas was being analysed. Even in the case of CO, where statistical approaches achieved comparable scores, all ML methods still performed better.

ML algorithms not only outperformed their statistical counterparts, but also displayed more consistent results across all gas types, with lower standard deviations. Overall, all ML models performed very similarly, with MLP being less consistent, with a higher standard deviation due to a slightly poorer performance on NO_2.

The last column from Table 3 shows the multitask results – training a single model for all gas types – with only the average over all gas types being shown. Random Forest displays the same performance as observed when training individual models, while we observe a small drop in performance for KNN and a significant drop for MLP.

4.1. Contextual Information

To measure the value of adding contextual information (Temperature, in our case), we show the results of training the models with different input attributes (Table 4). The statistical methods do not have a straightforward way to use additional information, therefore some results are omitted.

Table 4. Comparison of classification AUC of the presented algorithms using different input attributes.

Classifier	Reading	Temperature	Reading & Temp
Mean	0.89	0.54	N/A
Median	0.93	0.56	N/A
MLP	0.93	0.68	0.93
KNN	**0.94**	0.66	0.96
RF	0.93	**0.71**	**0.97**

Using Temperature alone leads to a very poor performance, as expected. Nonetheless, all ML models performed better than random choice. We believe this is due to the Temperature sensor also presenting faulty behavior that is correlated to the sensors' main reading – e.g. when a pod as a whole is subject to damaging situations, all sensors within are affected.

Using gas concentration only, all ML models still perform better than the statistical counterparts. Performances between ML models in this case are again very similar, with a slight advantage for KNN. It is clear that Random Forest made the best use of the extra data, showing the largest improvement when adding Temperature information.

Temperature is known to influence a sensor's readings and this influence is part of specification of any given sensor [Mead et al. 2013]. However, the performance gains observed when adding temperature as an input show that there are significant correlations between temperature and failures. Nonetheless, the statistical methods, trained only on Temperature, were largely unable to make use of this correlation, producing results close to random chance.

These results show that the gains obtained by ML models are both from being able to build more complex boundaries between examples and using extra contextual information that may have a correlation with faulty readings.

4.2. Dataset Size

To investigate how the algorithms perform given limited data, we evaluated the AUC by varying the dataset size available for training.

The results are shown in Fig. 2. As it can be observed, classifiers' ranking is somewhat stable except for very small data samples. Diminishing returns are observed after about 500 data points, but both Random Forest and KNN are still showing improvements even after the full dataset was made available, suggesting that larger datasets could further improve the observed performances for these classifiers.

On the other hand, Mean, Median and MLP show no improvement beyond a certain size. Moreover, MLP shows a high variance across sizes and is the only ML model to underperform the statistical methods for smaller datasets – neural networks in general are know to require a large amount of training data [Alwosheel et al. 2018], although Fig. 2 shows no evidence that the MLP would benefit from larger samples.

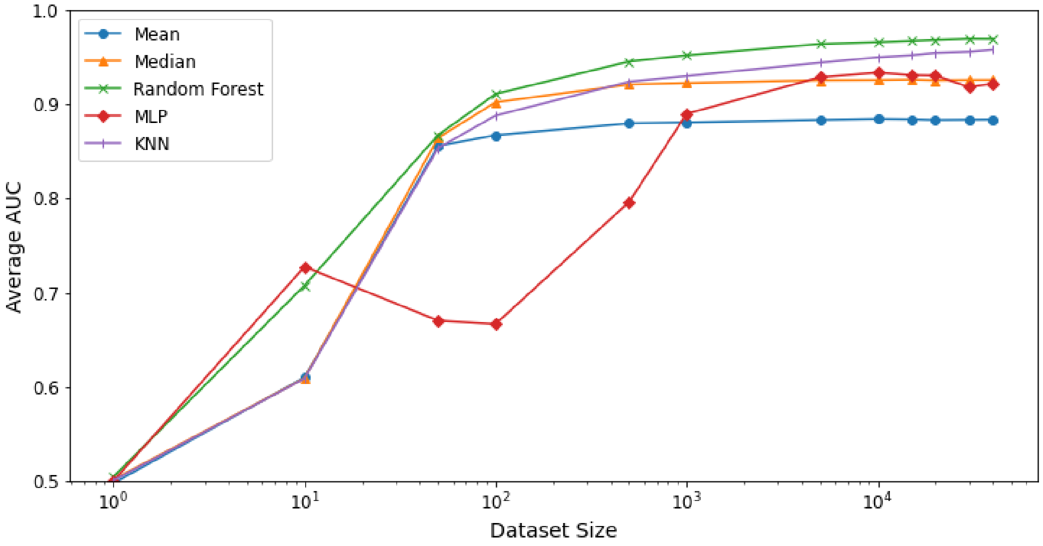

Figure 2. Average AUC for different dataset sizes.

5. Conclusions

In this paper we conducted extensive experiments on the application of machine learning models to the problem of detecting faulty readings in indoor air quality sensors. Leveraging a large dataset of 56,605 labeled readings from gas concentration sensors, we compared a Multilayer Perceptron, K-Neareast Neighbors and Random Forests models to two common statistical approaches, one of them currently being used for fault detection on data from the sensors used here.

We showed that all ML methods outperform their statistical counterparts, both when only using sensor readings and when adding contextual information in the form of the temperature at the time of the reading. The best ML provided a 4% gain on AUC over the best statistical method and 9% over the currently implemented method. Tying to our first specific goal of the paper we conclude that ML methods can perform fault detection significantly better than standard statistical approaches.

When allowed to use temperature data, ML models, in particular Random Forests, were able to use this extra information to deliver improved performance - up to about 4% gain was observed. Regarding our second specific goal, we conclude that adding contextual information is indeed beneficial and the seamless way that ML models integrate this information is a major benefit of such models.

Finally, we showed that the amount of data available for training impacts, as expected, the performance of all classifiers. For all of them, diminishing returns were observed after about 500 data points are used. KNN and RF outperformed statistical methods for all tested sizes, while MLP required larger datasets to achieve the same result. Notably, we provided evidence that both KNN and RF could benefit from even larger datasets than we had available, while MLP did not show such a tendency. Tying to our third and last specific goal of the paper we conclude that the evaluated ML models require approximately 1,000 data points to perform adequately but that larger amounts can lead to better performances, albeit diminishing ones.

Throughout the paper we made an attempt not to fine-tune the models to the problem, in particular regarding hyper-parameters tuning, so as not to overfit the models to the data. It is expected that by doing so, improved performance can be attained, in particular for neural networks. However, the observed performances were already close to perfect classification, making further efforts to improve performance less enticing. Nonetheless, future work includes fine-tuning the models to the problem, along with testing other more complex models, such as deep neural networks (e.g. [Eren 2017]).

Another line of future work is to compare point-wise detection as done here, where a single data point must be classified, to temporal classification, where a series of data points is passed to the model (e.g. [Loy-Benitez et al. 2020] and [Gupta et al. 2020]). The latter requires larger models and can introduce lag to the detection, which is a trade-off that can impact its usefulness in embedded and real-time settings.

References

Alwosheel, A., van Cranenburgh, S., and Chorus, C. G. (2018). Is your dataset big enough? sample size requirements when using artificial neural networks for discrete choice analysis. *Journal of choice modelling*, 28:167–182.

Blum, A., Kalai, A., and Langford, J. (1999). Beating the hold-out: Bounds for k-fold and progressive cross-validation. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 203–208.

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159.

Eren, L. (2017). Bearing fault detection by one-dimensional convolutional neural networks. *Mathematical Problems in Engineering*, 2017.

Friswell, M. I. and Inman, D. J. (1999). Sensor validation for smart structures. *Journal of intelligent material systems and structures*, 10(12):973–982.

Goldberger, J., Hinton, G. E., Roweis, S. T., and Salakhutdinov, R. R. (2005). Neighbourhood components analysis. In *Advances in neural information processing systems*, pages 513–520.

Gupta, S., Chatar, C., R Celaya, J., et al. (2020). Recurrent auto encoders for automatic sensor validation; tomorrows data with yesterday's sensors. In *IADC/SPE International Drilling Conference and Exhibition*. Society of Petroleum Engineers.

Henry, M. and Clarke, D. (1993). The self-validating sensor: rationale, definitions and examples. *Control Engineering Practice*, 1(4):585–610.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE.

Ibarguengoytia, P. H., Sucar, L. E., and Vadera, S. (2001). Real time intelligent sensor validation. *IEEE Transactions on Power Systems*, 16(4):770–775.

Kerschen, G., De Boe, P., Golinval, J.-C., and Worden, K. (2004). Sensor validation using principal component analysis. *Smart materials and structures*, 14(1):36.

Kubat, M. (1999). Neural networks: a comprehensive foundation by simon haykin, macmillan, 1994, isbn 0-02-352781-7. *The Knowledge Engineering Review*, 13(4):409–412.

Loy-Benitez, J., Heo, S., and Yoo, C. (2020). Soft sensor validation for monitoring and resilient control of sequential subway indoor air quality through memory-gated recurrent neural networks-based autoencoders. *Control Engineering Practice*, 97:104330.

Mattern, D., Jaw, L., Guo, T.-H., Graham, R., and McCoy, W. (1998). Using neural networks for sensor validation. In *34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, page 3547.

Mead, M., Popoola, O., Stewart, G., Landshoff, P., Calleja, M., Hayes, M., Baldovi, J., McLeod, M., Hodgson, T., Dicks, J., et al. (2013). The use of electrochemical sensors for monitoring urban air quality in low-cost, high-density networks. *Atmospheric Environment*, 70:186–203.

Muller, C., Fish, P., Glover, N., McElligott, R., and Bennett, D. (2018). Does control of indoor co2 levels negatively impact iaq? In *Does Control of Indoor CO2 Levels Negatively Impact IAQ?*

Napolitano, M. R., Windon, D. A., Casanova, J. L., Innocenti, M., and Silvestri, G. (1998). Kalman filters and neural-network schemes for sensor validation in flight control systems. *IEEE transactions on control systems technology*, 6(5):596–611.

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

Upadhyaya, B. R. and Eryurek, E. (1992). Application of neural networks for sensor validation and plant monitoring. *Nuclear Technology*, 97(2):170–176.

World Health Organization (2020). Air pollution. `https://www.who.int/westernpacific/health-topics/air-pollution`. Accessed: 2020-03-20.

Yang, J., Sun, Z., and Chen, Y. (2016). Fault detection using the clustering-knn rule for gas sensor arrays. *Sensors*, 16(12):2069.