

Indoor Air Pollution Forecasting using Deep Neural Networks



Jorge Altamirano-Astorga,
Luz Aurora Hernández-Martínez,
Ita-Andehui Santiago Castillejos
Edgar Román-Rangel

Instituto Tecnológico Autónomo de México

Introduction.






Indoor air quality is an increasing concern because a large portion of the population stays inside indoor spaces most of the time in places such as: schools, offices and at home. Because of COVID-19 pandemic this issue has also gathered attention.

We found that although there is previous work, as described in the paper: there's a lack of published open source models and not many papers detailing forecasting indoor pollution.

Our paper tries to address:

1. Forecasting indoor air quality can have a decision making impact.
2. To advance further research through an open research, open source and open data approach.







Data: Sources.

Source	Description	Records	Granularity
	Indoor Air Quality Sensor: Bosch BME680.	+6.2 M	Every 3 seconds recorded in a CSV in a RaspberryPi.
	SINAICA: Air Quality Monitoring Stations of the Government.	+2.3 k	Every 1 hour downloaded from the SINAICA Government Open Data website in several CSV files.
	OpenWeatherMap: Weather Data.	+5 k	Every 1 hour. Data paid and downloaded from OpenWeatherMap in a CSV file.

Data: Variables Used for Training the Models.

Dimensions Used for Training:

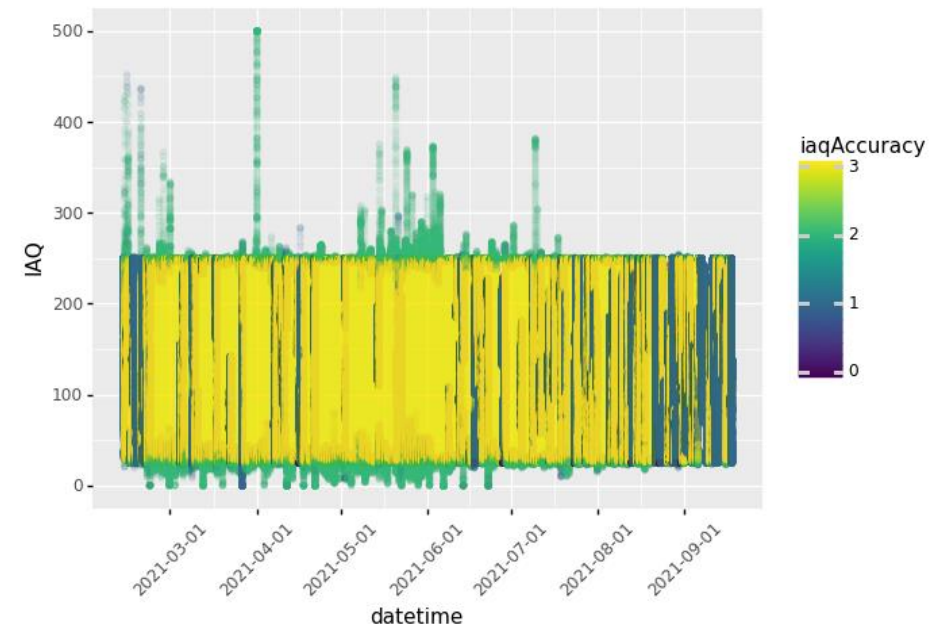
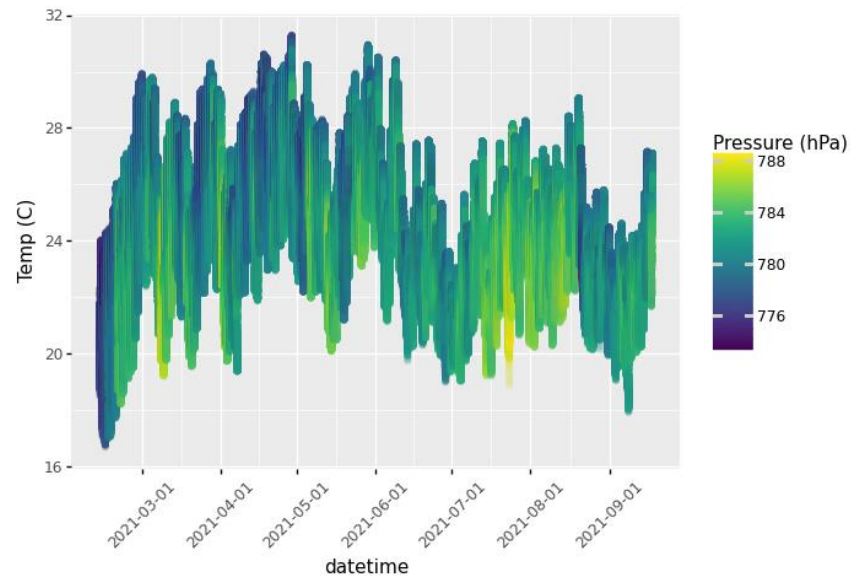
- i. 15 variables (features).*
- ii. 43906 records.*
- iii. Window history size.*

Sources	Variables	Values	Type of Variable
	Temperature	-40C – 85C	Continuous
	Humidity	10% – 95%	Continuous
	Atm Pressure	300 hPa – 1100 hPa	Continuous
	Date	2/2/2021 – 9/27/2022	*
	Pollulants: incl. CO, NO, NO2, O3, PM2.5, PM10	ppm mainly, but may include ppb.	Discrete
	IAQ	0 IAQ - 500 IAQ	Discrete

\mathcal{X}

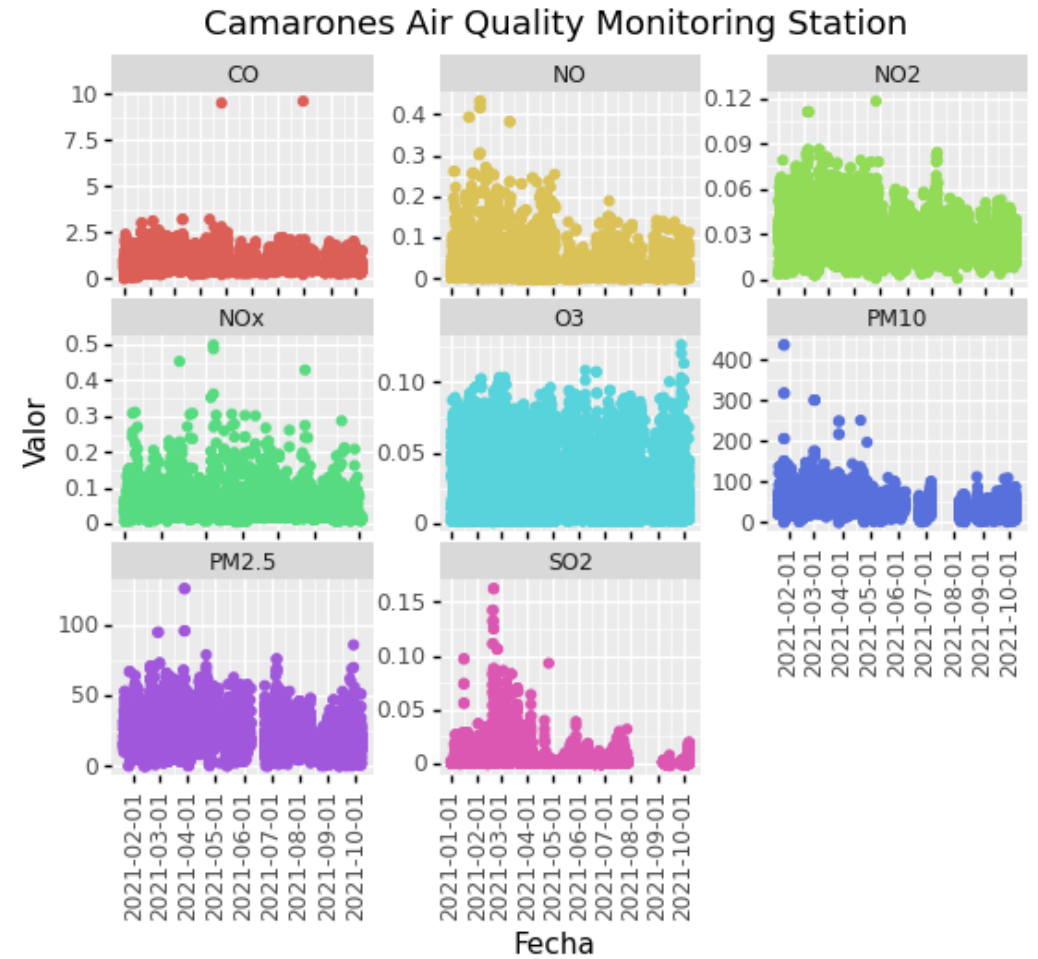
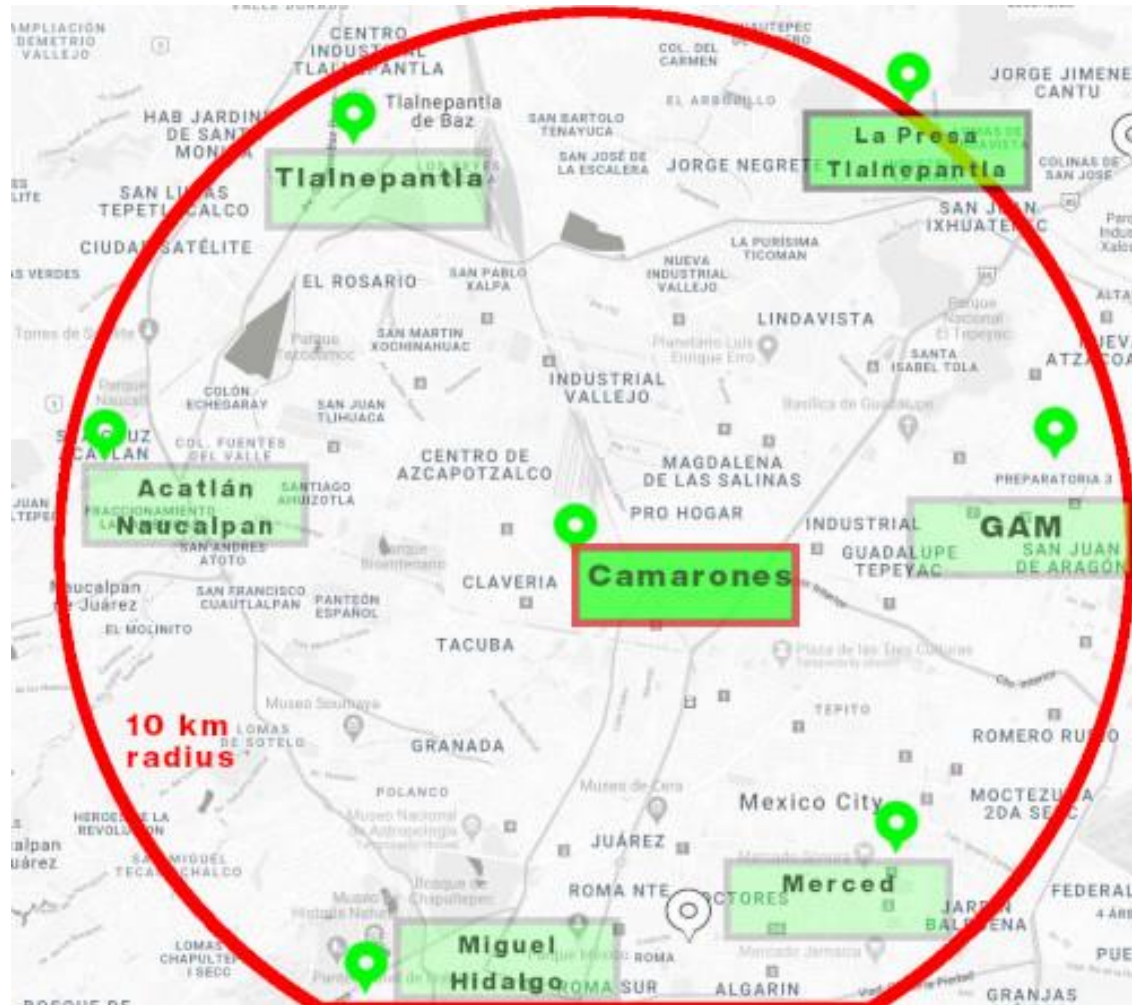
\mathcal{Y}

Exploratory Data Analysis: of the Sensor.



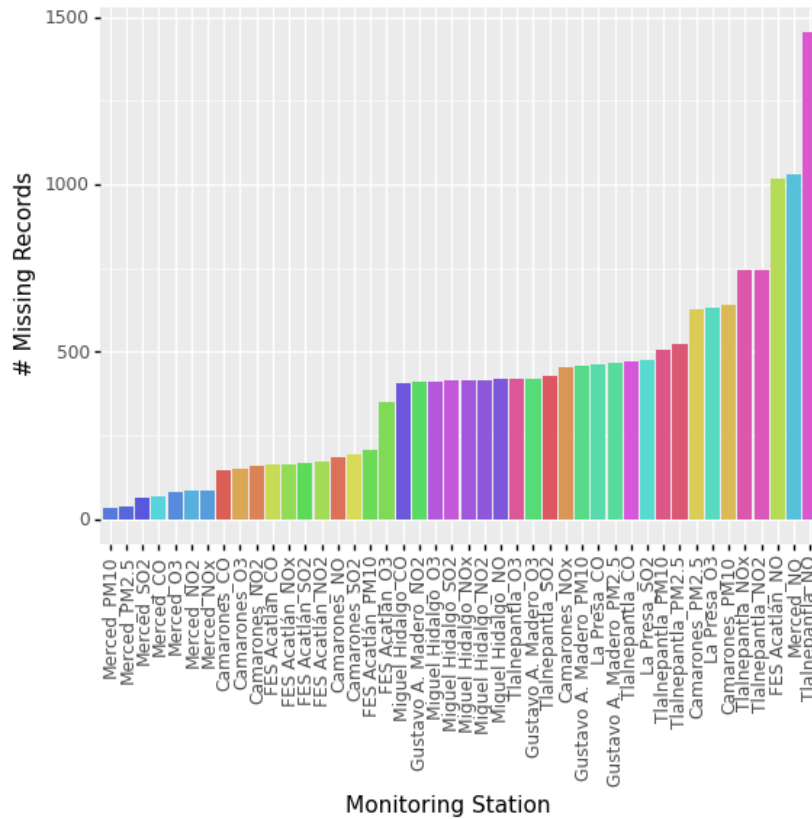
Missing Data: ~1%

Exploratory Data Analysis: Government Data (SINAICA).

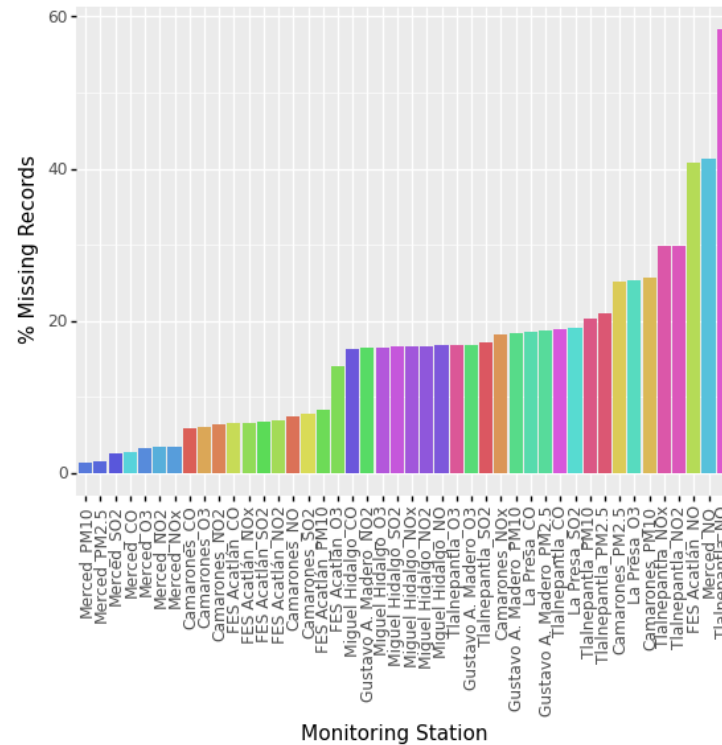


Exploratory Data Analysis: Government Data (SINAICA).

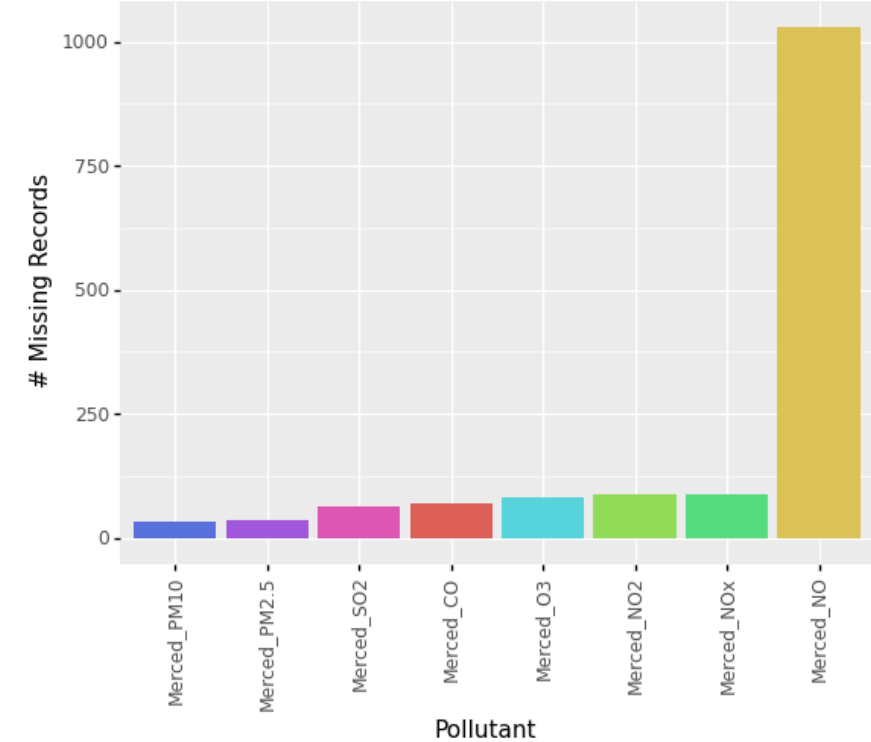
Missing Records Histogram by Pollutant and Government Station



Percentages of Missing Records by Pollutant and Monitoring Station

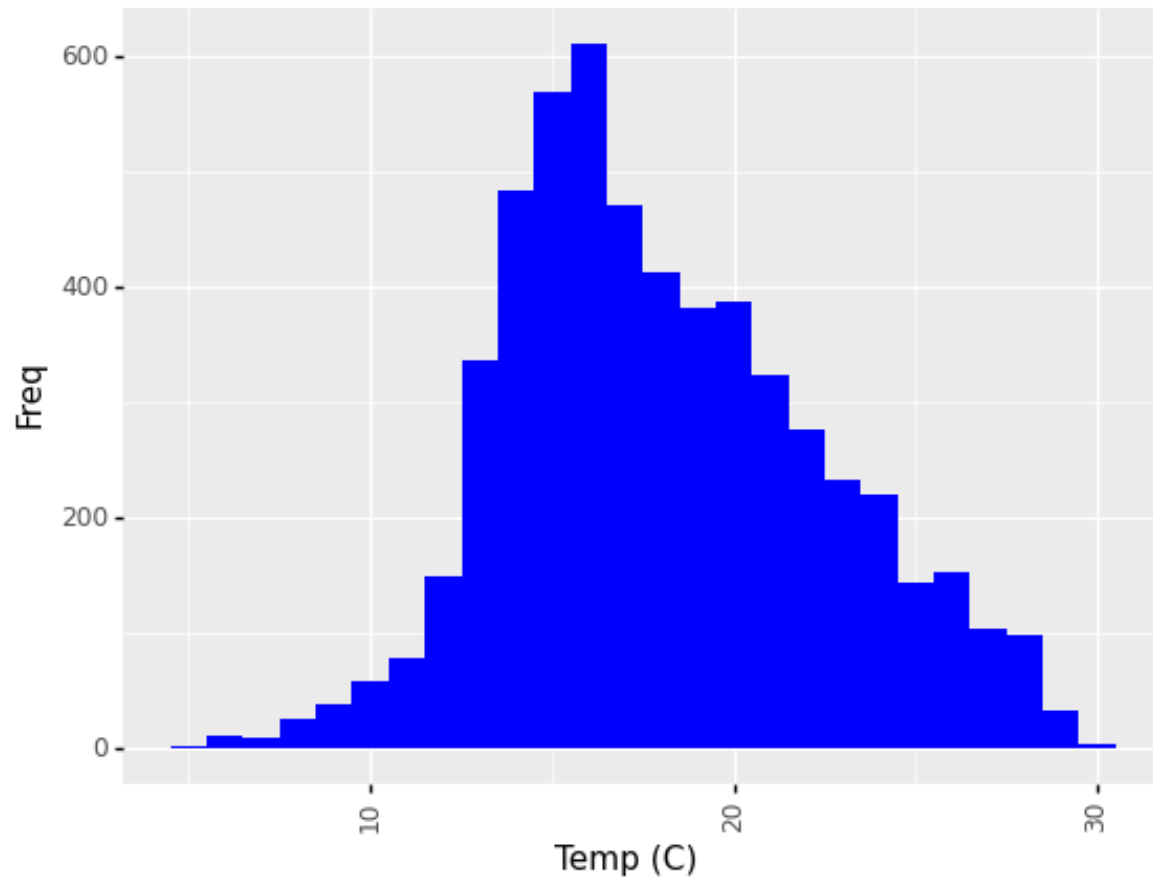


Histogram of Missing Records by Pollutant in La Merced

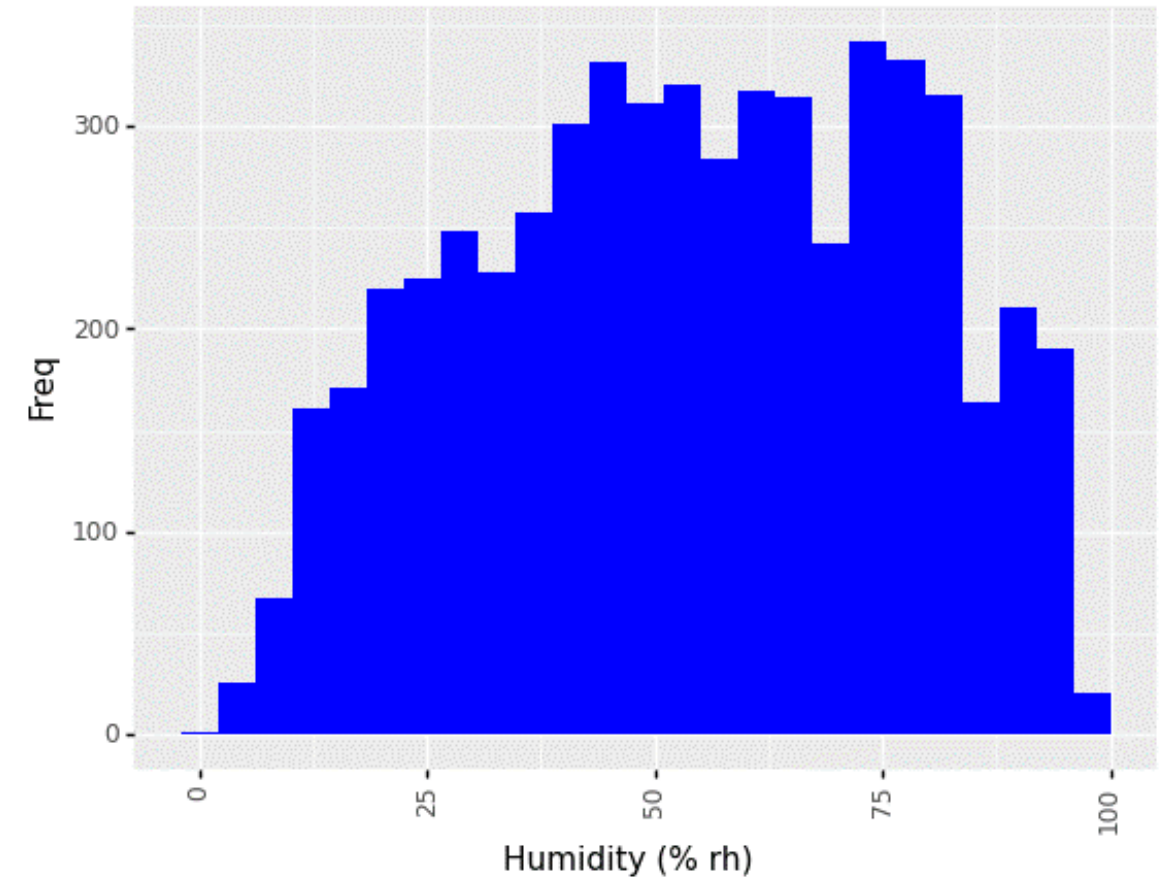


Exploratory Data Analysis: Weather Data.

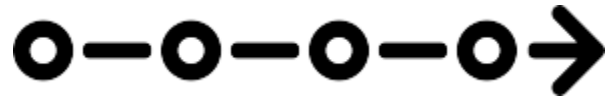
Histogram Plot of Weather Conditions: Temperature



Histogram Plot of Weather Conditions: Humidity



Implementation: Data Preprocessing.



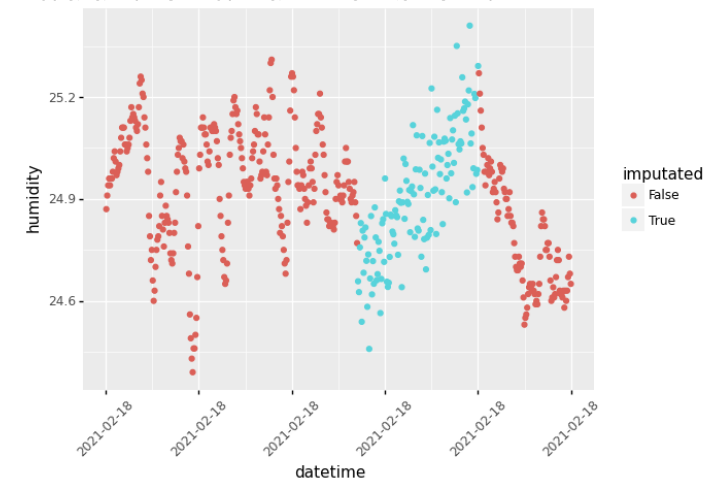
We treated the data as a timeseries for all processing, based on previous work documented on the Tensorflow / Keras timeseries tutorial. Aggregating all datasources into a single dataframe.



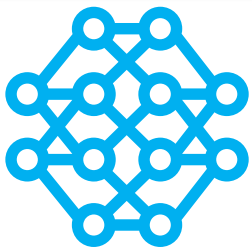
Both the sensor data and the government data had missing records. We explored several data imputation approaches (KNN, GLM, Bayesian GLM). We found the best approach was with linear interpolation. Adding some noise as shown in this figure to avoid overfitting on our DL models.

Scalers: we did use MinMaxScaler as the different variables were in very different scales.

Time-series Processing: We created history windows transforming matrices (dataframes) to tensors using history windows as an additional dimension.

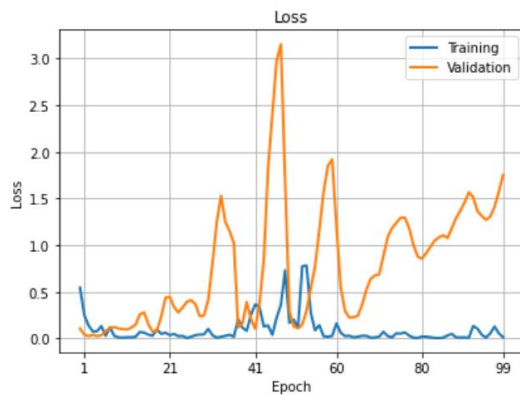


Implementation: Neural Nets Architecture.



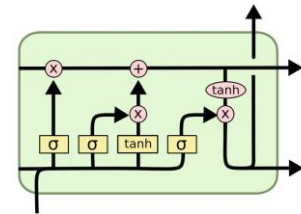
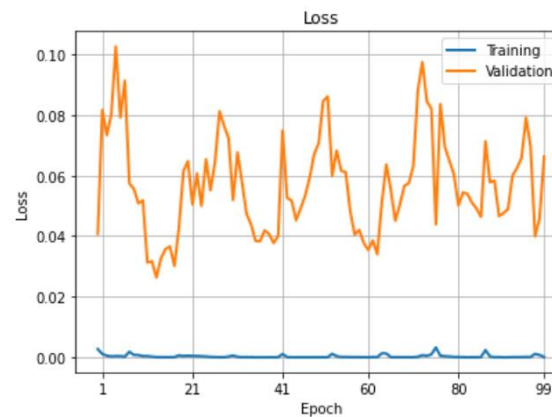
Perceptron:

- *Fast training.*
- *Competitive performance.*
- *This is the base of all artificial neuron types.*



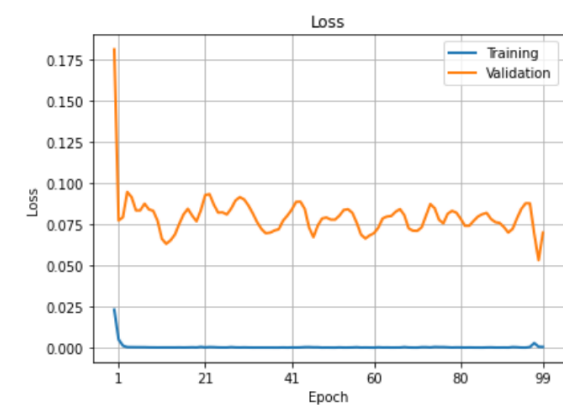
Convolutional:

- *Usually good performance.*
- *High processing resources.*
- *Noisy and more unreliable results.*

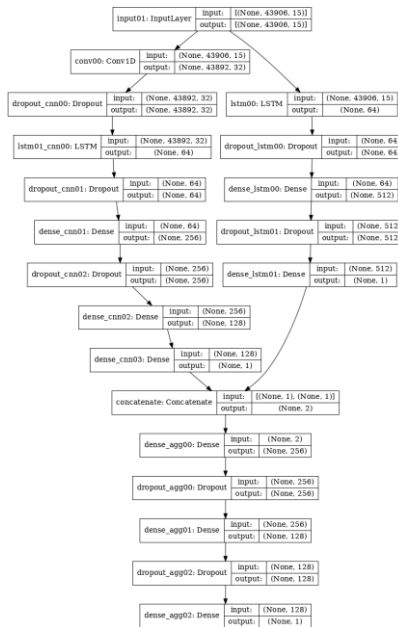
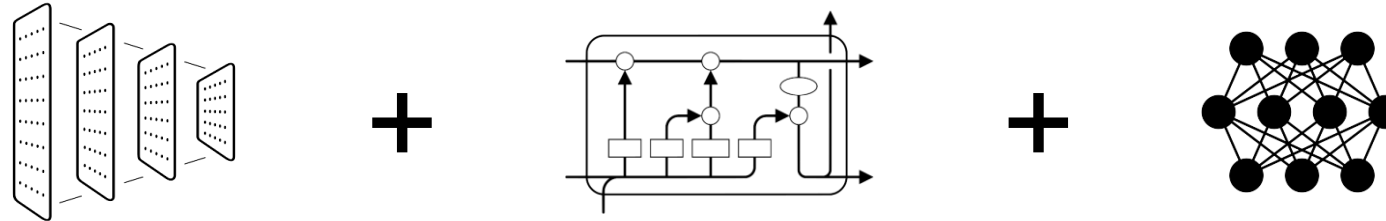


LSTM:

- *Robust performance.*
- *High on memory resources.*
- *More reliable performance.*

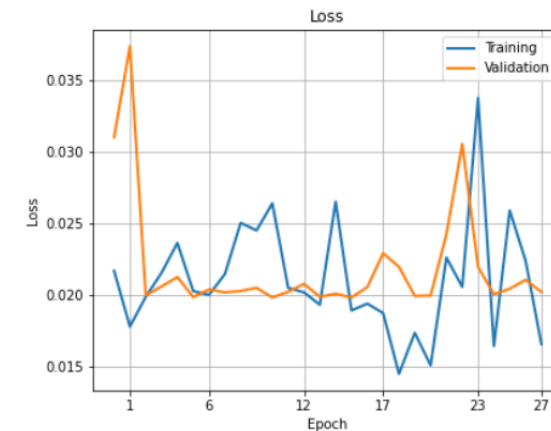


Implementation & Results: Combination of Architectures into a Single Model.



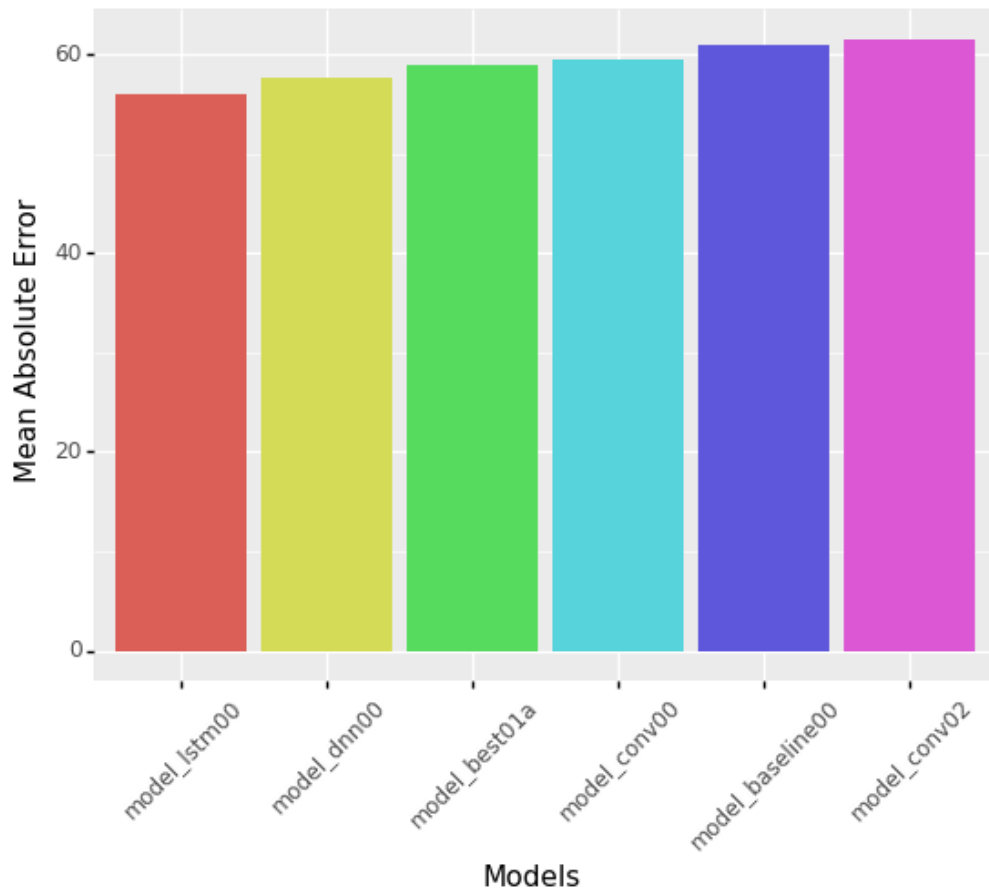
- *We expected better results.*
- *Reliable performance.*
- *Reasonable training resource usage.*
- *More complex to code, but attainable: helped to flex our coding muscles.*

Model	Time	Epochs	W. Sz.	Stride	Samp. Rt.	MSE	MAE	Re Sample
lstm00	47.62	11	30	2	2	0.0249	55.97	10 Min
dnn00	29.58	14	15	1	2	0.0309	57.58	10 Min
best01a	890.75	28	15	1	2	0.0244	59.03	05 Min
conv00	79.88	14	15	1	2	0.0301	60.34	10 Min
baseline00	24.97	14	15	1	2	0.0325	60.90	10 Min
conv02	799.29	28	15	1	2	0.0207	61.58	05 Min



Results.

Plot for Comparing the Models Performance on IAQ Scale.



We found that the results favored:

- LSTM with 4 weeks of history: the more history the model has available, performance improves. LSTM seems to be doing its job.
- LSTM performed consistently well on our tests.
- LSTM performed better with stride 2, but the other models performed better with stride 1.
- Perceptron networks (DNN00) performed surprisingly well.
- We expected a better performance with a more complex neural network architecture combining different NN types.
- Our tests favored simpler NN architectures.
- Surprisingly, convolutional networks didn't performed well, worse than a baseline model, i.e. no activation single perceptron architecture.

Conclusions and Further Research.

- Conclusions:
 - We combined several data sources and compared different models' performance successfully.
 - We discovered that it is possible to improve the performance combining different data engineering techniques.
 - We leveraged the cloud computing power in a replicable fashion for hyperparameter tuning and model comparison with better results than investing in complex architecture models.
- Further research:
 - Exploring the models learning capabilities by using more data.
 - Using a transfer learning approach for timeseries data.
 - Creating a larger model and feeding it with more historic data.

Thank you!

Q & A



Image Credits.

- File:LSTM.jpg. (2020, October 12). Wikimedia Commons, the free media repository. User MingxianLin. Retrieved May 20, 2022 from <https://commons.wikimedia.org/w/index.php?title=File:LSTM.jpg&oldid=487942515>.
- Home (2022, May 12). The Noun Project. User Vectorstall. Retrieved May 20, 2022 from <https://thenounproject.com/icon/home-4864927/>.
- Air Pollution (2020, July 24). The Noun Project. User Eucalyp. Retrieved May 20, 2022 from <https://thenounproject.com/icon/air-pollution-3631564/>.
- Mexico City (2020, January 23). The Noun Project. User Blaise Sewell. Retrieved May 20, 2022 from <https://thenounproject.com/icon/mexico-city-3583774/>.
- Convolutional Neural Network (2019, July 19). The Noun Project. User Oleksandr Panasovsky. Retrieved May 20, 2022 from <https://thenounproject.com/icon/convolutional-neural-network-2863992/>.
- Deep Learning (2021, July 18). The Noun Project. User Mohamed Mb. Retrieved May 20, 2022 from <https://thenounproject.com/icon/deep-learning-4321517/>.