Proyecto Final

, , ,

May 6, 2021

1 *Proyecto Final*

Jorge III Altamirano Astorga, Luz Aurora Hernández Martínez, Ita-Andehui Santiago Castillejos.

Mayo 7, 2021

2 Resumen

Desarrollaremos un proyecto de investigación basados en un sensor de la calidad del aire que tenemos dentro de casa de uno de los participantes con el fin de estudiar, analizar, explorar y entender su relación e influencia con los fenómenos externos (calidad del aire de la ciudad y variables atmosféricas) para poder predecir la calidad del aire en el interior de casa con las mediciones de compuestos orgánicos volátiles, los cuales tienen alta probabilidad de ser perjudiciales para la salud.

3 Introducción

El ITAM y muchos de sus miembros tenemos residencia en la Ciudad de México. Esta metrópolis es una de las más contaminadas en el continente y en el mundo. A causa de la pandemia, una gran cantidad de la población pasamos mucho de nuestro tiempo encerrados en espacios cerrados, típicamente nuestras viviendas. Queremos saber cómo influyen los factores atmosféricos y la contaminación de la Ciudad en la contaminación de un espacio cerrado.

Hemos almacenado los datos de este sensor desde Febrero 2021 tratando de mantenerlos en un área común que no tiene ventilación directa para evitar perturbaciones en las lecturas y que sea influído directamente por la contaminación exterior, así como de otras fuentes de emisión (cocina).

Es importante destacar que este sensor no detecta contaminantes primarios, tales como: óxidos de nitrógeno (NOx), dióxido de carbono (CO2), monóxido de carbono (CO), ozono (O3), más bien mide los compuestos orgánicos volátiles, conocidos por el acrónimo anglosajón VOCs. Los VOCs típicamente son muchos de los olores que percibimos, los cuales son disoluciones de compuestos en el aire.

Estos compuestos orgánicos volátiles se han comprobado como nocivos a la salud y posibles cancerígenos, lo cual nos despertó el interés. Ejemplos de estos compuestos orgánicos son: el humo del cigarro, humo causado por cocinar alimentos, la utilización de agentes de limpieza (cloro y basados en amoniaco) y fuentes volátiles varias (como solventes, pinturas, quitaesmaltes), entre otros.

3.1 Fuentes de Datos Prospecto

Tenemos los siguientes de fuentes de datos:

- Sensor Bosch BME680: contamos aproximadamente con casi 1.3 millones de registros con lecturas del sensor cada 3 segundos.
- Datos Abiertos de la Calidad del Aire del Gobierno de la Ciudad de México: datos por hora de las estaciones de monitoreo del Gobierno.
- Datos de otras estaciones meteorológicas de la Ciudad.

3.1.1 Problemáticas

- Datos del Gobierno de la Ciudad de México: estos datos no están siendo actualizados de manera cotidiana, por lo que en ocasiones tienen atrasos en publicar la información actual. Ejemplo, al 31 de marzo no habían subido actualizaciones desde el 28 de enero. Pensamos contrarrestarlo utilizando la fuente de datos federal (Sinaica) que se encuentra en la bibliografía.
- Datos meteorológicos de terceros: Pudieran ser fuentes propietarias que serían de difícil acceso o hubiera que utilizar minería de datos (webscraping). Continuaremos buscando fuentes abiertas y confiables.
- Precisión y manipulación de los datos de nuestras fuentes de datos.
- Estabilidad y precisión de la toma de registros en el sensor. Pudiéramos tener interrupciones del suministro eléctrico que no permitieran obtener ciertas lecturas.
- Algoritmo cerrado del sensor para convertir de la variable `gasResistance` a la variable `IAQ`; el cual es cerrado. Sólo proporcionan un objeto binario, que se puede ejecutar en Linux, Windows, ARM y ligándolo en C, el cual procesa la variable `gasResistance`. [1] [2]
  - Al parecer esta variable se genera como una serie de tiempo y llega a guardar hasta 24 hrs de estados previos de ejecución del sensor. [3]
  - Este código es protegido como propiedad intelectual del fabricante del sensor (Bosch), lo cual impide que se pueda hacer desensamble o ingeniería inversa.
  - Se pudiera utilizar el objeto binario en un ejecutable en Linux o Windows, donde pudiese ser convertida no desde el sensor, sino desde nuestras predicciones de la variable gasResistance; y con este código binario más nuestro ejecutable que ligue ambos convierta los valores de gasResistance.
  - Alternativamente se puede generar una red neuronal que convierta la variable `gasResistance` (como variable explicativa, incluso agregar las otras) y convertirlas a IAQ (variable respuesta).

3.2 Variables

Las siguientes son las variables obtenidas por el sensor Bosch BME680:

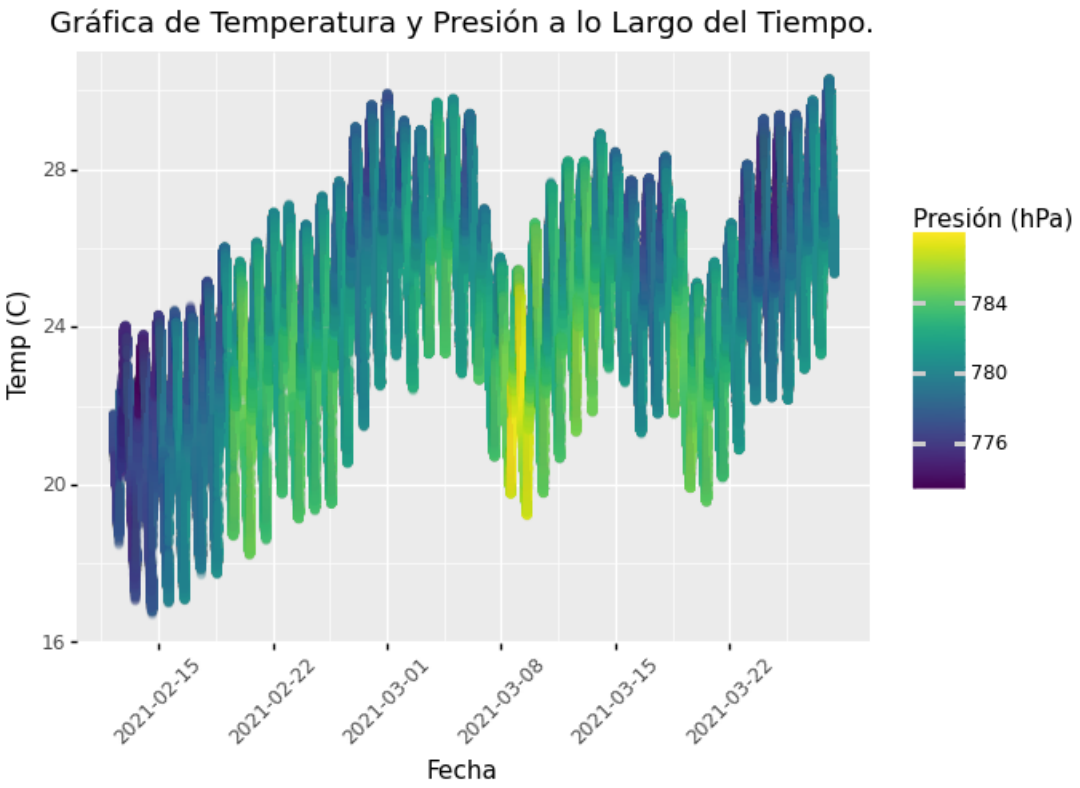
- Temperatura: variable numérica en grados Celsius (C) con una resolución de $0.01C$ y una precisión de $\pm 0.5C$.
- Presión: variable numérica en hectopascales (hPa) con una resolución de 0.18 hPa y una precisión de ± 0.12 hPA.

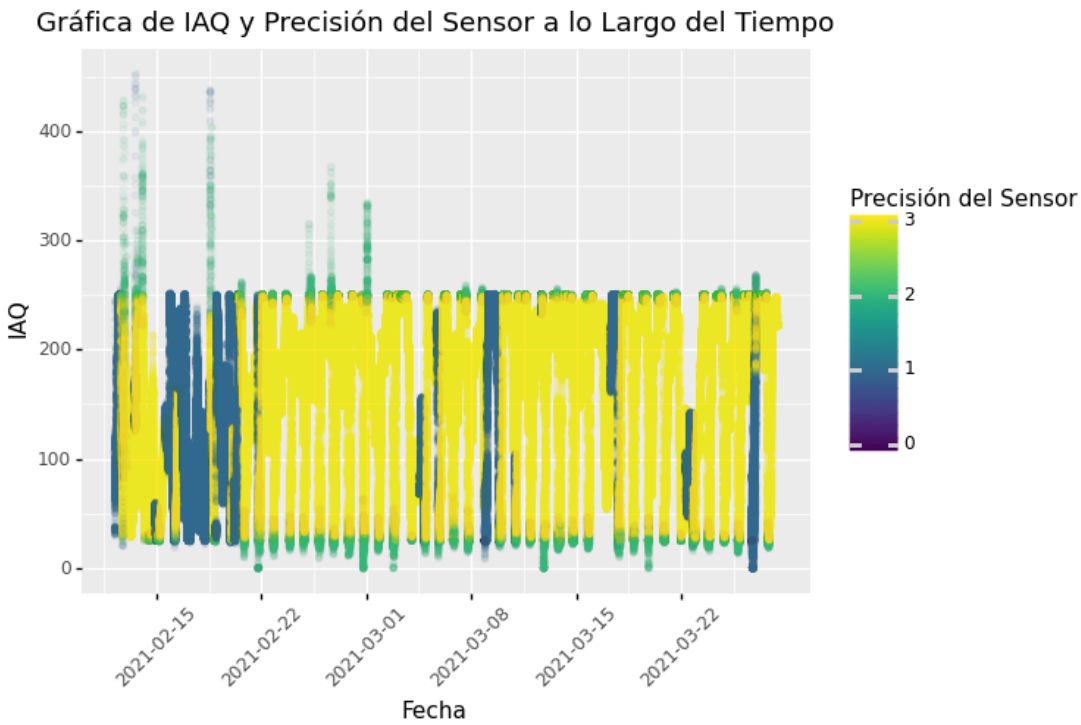
- Humedad: variable numérica en porcentaje de humedad relativa (%rH) con una resolución de 0.008 $\%rH$ y una precisión de $\pm 3\%rH$.
- Resistencia del Gas: variable numérica de la resistencia eléctrica opuesta al elemento sensible del sensor medida en Ohms.
- IAQ: variable numérica medida en el índice de calidad del aire americano en interior (IAQI, aunque utilizaremos la nomenclatura IAQ) con una resolución de 1 IAQ. La precisión del sensor variable que no excede 5% se guarda en una variable independiente.
- Precisión del sensor: variable categórica ordinal con valores en el rango de [0,3]:
  - 0: periodo de estabilización o no operativo.
  - 1-2: periodo operativo.
  - 3: precisión máxima y operación óptima.
- Fecha y hora: variable numérica basado en UNIX/POSIX epoch que denota el tiempo desde el 01/01/1970 00:00:00.0 UTC. El tiempo está sincronizado por NTP al Centro Nacional de Metrología de México (Hora Oficial del País).

3.3 Análisis Exploratorio Inicial

Hemos empezado a realizar un análisis exploratorio de los datos y recopilación de los datos de fuentes externas. A continuación mostramos algunas gráficas de las lecturas del sensor.

- Rango de fechas obtenidas: 2021-02-12 06:04:09.089621067 - 2021-03-28 10:23:00.196267605
- Número de registros: 1,274,818
- Promedio de IAQ: 159.75 desviación estándar: 73.65


Gráfica de Temperatura y Presión a lo Largo del Tiempo.
Temp (C)
28
24
20
16
2021-02-15
2021-02-22
2021-03-01
2021-03-08
2021-03-15
2021-03-22
Fecha
Presión (hPa)
784
780
776


Gráfica de IAQ y Precisión del Sensor a lo Largo del Tiempo
IAQ
400
300
200
100
0
2021-02-15
2021-02-22
2021-03-01
2021-03-08
2021-03-15
2021-03-22
Fecha
Precisión del Sensor
3
2
1
0

3.4 Modelo

Proponemos utilizar un modelo de aprendizaje profundo con distintos tipos de neuronas artificiales:

- RNN: red neuronal recurrente.
- CNN: red neuronal convolucional.
- DNN: red neuronal densa.

Además de utilizar técnicas de series de tiempo, estadística frecuentista y bayesiana para el análisis de los datos.

4 Trabajos relacionados

Hemos realizado algunos trabajos previos (1) y buscado artículos relacionados que describimos a continuación:

1. Examen final para la materia de "Modelos de Gran Escala" con la Prof. Liliana Millán, donde se estudiaron la relación de las estaciones de biciletas "Ecobici" con la calidad del aire en las inmediaciones.
2. Development of indoor environmental index: Air quality index and thermal comfort index. Referido en la bibliografía.

5 Solución

Hicimos un procesamiento de los datos imputando las observaciones faltantes con datos de la calidad del aire de la CDMX, en específico con la información del monitor de la estación Camarones, esto considerando su distancia a la ubicación de nuestro sensor. Para hacer la imputación usamos interpolación, aunque también exploramos KNN, MLG, entre otros. También es importante mencionar que procesamos los datos como una serie de tiempo usando un tutorial de tensorflow y Keras [5].

El modelo que mejor resultados nos dio fue un modelo de redes neuronales convolucionales (CNN-1D), el beneficio de usarlo para la clasificación de secuencias es que pueden aprender directamente de los datos de series de tiempo sin procesar y, a su vez, no requieren experiencia en el dominio para diseñar manualmente las características de entrada. El modelo aprendió una representación interna de los datos de la serie temporal y logró el mejor rendimiento.

También probamos con los siguientes modelos: * Redes neuronales recurrentes (RNN), que son la arquitectura más clásica y utilizada para problemas de predicción de series temporales; * Long Short-Term Memory (LSTM), que son una evolución de las RNN desarrolladas para superar el problema del gradiente que desaparece; * Redes neuronales densas (DNN) Una red neuronal profunda (DNN) es una red neuronal artificial (ANN) con múltiples capas entre las capas de entrada y salida; * Mezcla de los mejores modelos, en este caso CNN + LSTM.

#INSERTAR TABLA COMPARATIVA

6 Conclusiones

Buscaremos explicar nuestras conclusiones del análisis de:

- Análisis exploratorios sobre los datos.
- Nuestras hipótesis y sus resultados.
- Nuestras pruebas con modelos de redes neuronales profundas y el procesamiento de señales.
- Los inconvenientes que encontramos y la influencia que ejercieron sobre el presente proyecto.
  - Que no tuviéramos todos los datos disponibles en el sitio de la SINAICA nos complicó muchísimo no poder tener más datos.
  - El algoritmo propietario (cerrado) para calcular la variable IAQ a partir del sensor de la resistencia eléctrica en el aire hizo que perdiéramos tiempo en probar cuál de las variables era más viable predecir.
  - Aunque *a caballo regalado no se le ve el diente*: Los tiempos y capacidades de Google Colab no son los mejores, aún con Pro (que es bastante económico). Además de que su velocidad de desempeño es aleatoria durante diferentes momentos.
  - Las fuentes externas no siempre son confiables, gratuitas o completas. Nos hubiera gustado tener una fuente de datos de clima gratuita y que fuera confiable.
- Aprendizajes obtenidos durante el presente.
  - Nunca se debe subestimar la necesidad (y el tiempo que toma) de limpiar, explorar, imputar, "corregir" los datos.
  - Cualquier mejora en tiempos conlleva un reto, a veces grande; tal como vimos en los miniproyectos y se comentó en clase.
  - No podemos confiar en que el gobierno o un ente externo entregue los datos a tiempo. Entendimos que es importante tener desde la fundación de un proyecto al menos con el 70% de los datos disponibles, para empezar a trabajar con ellos.
  - Nunca subestimar las fuentes bibliográficas oficiales (en nuestro caso Keras y Tensorflow) como guías prácticas y bien escritas. Dado que muchas veces Stackoverlow no tiene las mejroes respuestas. [4], [5]
  - No tener miedo a aprender y a leer es vital para estar actualizado: pues muchas veces cambia mucho las APIs, nuevos "approaches".
  - Nos resultó muy útil tener un modelo baseline [5] que fue como una hipótesis nula, que nos permitió saber qué hacía un modelo vacío. Es bueno tener esta comparación y la rescatamos como valiosa.
  - Los detalles y el cuidado de estos son de suma importancia:

  i. Aprendimos que escalamos los datos, pero devolverlos a una escala real, es otra cosa.

  ii. Cuidar las semillas para hacer lo más reproducible es crítico.

  iii. En nuestro caso fue importante cuidar que ninguna manipulación (como el `train_test_split` no hiciera `shuffle` o barajeara el orden.

iv. Es importante probar, probar y probar: alterando 1 solo detalle de la arquitectura, un hiperparámetro, una modificación de los datos, etc. Esto involucra hacer modelos constructivos, no hacerlos complejos desde un inicio.

- Siguientes pasos que nos surgieron:

  i. Tomar datos por al menos 1 año, para establecer más claramente la estacionalidad, incorporar (comprar) fuentes de datos del clima y contaminantes, tener más puntos de redundancia (no depender únicamente de 1 sensor en 1 sólo lugar).

  ii. Explorar mejores opciones para imputar datos secuenciales o series de tiempo.

  iii. Realizar más pruebas sobre las predicciones y ver cómo van las mediciones.

  iv. Hacer las pruebas con mayor capacidad de cómputo: tener ventanas más grandes en `timeseries_dataset_from_array`. Que se nos ocurría extender o reescribir esta función de Keras promediando por ejemplo a 1 min, en vez de estar probando.

  v. Realizar más *Hyperparameter Tuning* (Learning Rate, combinar optimizadores en los modelos), buscar datos preentrenados, o entrenar la red con datos como el del ejemplo (del Departamento de Biogeoquímica del Instituto Max-Planck) [4] [5].

  vi. Hacer un tablero o que las predicciones sean predecibles por algún API.

  vii. Realizar predicciones con algún intervalo de confianza o grado de certidumbre de la predicción.

  viii. Mejorar el desempeño, que claramente no es el mejor.

  ix. Venderlo o publicarlo en algún journal.

7 Bibliografía

- [1] Bosch Sensortec Community | How do I convert BME 680 gas resistance to IAQ?.
- [2] Bosch SensorTec Comunity | Solution to IAQ accuracy definition.
- [3] GitHub | Daniel Mancuso: Código fuente de OhmTech-io/uThingVOC Src/main.c
- [4] Keras contributors et al. Keras / Code examples / Timeseries / Timeseries forecasting for weather prediction. 2021.
- [5] Tensorflow Contributors. Tensorflow: Tutorial on Time series forecastingTime series forecasting. 2021.
- Bosch BME680 Datasheet. 2021.
- Mancuso, Daniel. Indoor Air Quality Monitor | Hackster.io. 2019.
- Dirección de Monitoreo Atmosféerico de la Secretaría del Medio Ambiente del Gobierno de la Ciudad de México.
- Sistema Nacional de Información de la Calidad del Aire del Gobierno Federal México.
- Saadi S. M., et al. Development of indoor environmental index: Air quality index and thermal comfort index. 2017. doi:10.1063/1.4975276.

- Pedregosa, Fabian; et al. Scikit-learn: Machine Learning in Python. 2021.
- Abadi, Martín, et al. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015.
- McKinney, Wes. Data structures for statistical computing in python. 2010.
- Harris, Charles, et al. Array programming with NumPy. 2021. doi:10.1038/s41586-020-2649-2.
- Hunter, John. Matplotlib: A 2D Graphics Environment. 2007. doi:10.5281/zenodo.592536/zenodo.592536.
- Román-Rangel, Francisco. Notas y Código del Curso de Aprendizaje Profundo. 2021.
- González-Pérez, Felipe. Notas de aprendizaje de máquina. 2020.
- Mohd, Faizy. StackOverflow AI: How to use CNN for making predictions on non-image data?. 2021.
- Ondris, Ladislav. StackOverflow: How to apply CNN algorithm in python for non image dataset. 2020.
- Shady, Slim. Convolutional Neural Network With Tensorflow and Keras. 2021.
- Keras contributors et al. Keras API Reference: fit. 2021.
- Keras contributors et al. Keras API Reference: Convolution Layers > Convolution1D. 2021.
- Pedregosa, Fabian; et al. Scikit-learn: Machine Learning in Python. 2021.
- Abadi, Martín, et al. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015.
- McKinney, Wes. Data structures for statistical computing in python. 2010.
- Harris, Charles, et al. Array programming with NumPy. 2021. doi:10.1038/s41586-020-2649-2
- Hunter, John. Matplotlib: A 2D Graphics Environment. 2007. doi:10.5281/zenodo.592536/zenodo.592536