

## Propuesta de red neuronal convolutiva para la predicción de partículas contaminantes PM10

Ricardo Domínguez-Guevara, María del Carmen Cabrera-Hernández,  
Marco Antonio Aceves-Fernández, Jesús Carlos Pedraza-Ortega

Universidad Autónoma de Querétaro, Facultad de Ingeniería, Querétaro, México  
ricardo.dominguezguevara@gmail.com, k.armenkabrera@hotmail.com,  
marco.aceves@gmail.com, caryoko@yahoo.com

**Resumen.** La materia particulada o partículas suspendidas (PM) son partículas contaminantes que forman parte del aire que respiramos; un subgrupo de PM son las partículas con diámetro menor a 10 micrómetros, las cuales son nombradas PM 10. Este último subgrupo de partículas ha sido reconocido dentro del conjunto de contaminantes que mayor impacto tiene en la salud de las personas; son un factor importante en enfermedades cardiovasculares, que a su vez desarrollan otro tipo de padecimientos. Por esta razón, el poder contar con modelos que permitan determinar el comportamiento de PM 10 es fundamental para la prevención de algunas enfermedades. Por otra parte, en años recientes se ha explorado el uso de redes neuronales convolutivas (Convolutional Neural Networks o CNN) aplicadas a la predicción de series de tiempo, demostrando un gran potencial en dicho propósito. En el presente trabajo se implementa una arquitectura de CNN aplicada a la predicción de PM 10, utilizando distintas variables meteorológicas para el modelado de PM 10.

**Palabras clave:** red neuronal convolutiva, redes profundas, partículas contaminantes, PM 10, predicción, CNN.

### Convolutional Neural Network Proposal for Particulate Matter PM10 Prediction

**Abstract.** Particulate matter or suspended particles (PM) are pollutant particles that form the air we breathe; a subgroup of PM are the particles with a diameter lower than 10 micrometers, those particles are called PM 10. This subgroup has been recognized in the group of pollutants that has higher impact on people health; they are an important factor on cardiovascular diseases that also develop another health problem. For this reason, counting with models that let us determine the behavior of PM 10 is crucial for the prevention of some diseases. In the other hand, in recent years it has been explored the application of convolutional neural networks (CNN) on time series predictions, showing a huge potential for that purpose. In this work a CNN architecture is implemented and applied to the prediction of PM 10, using meteorological variables to model PM 10.

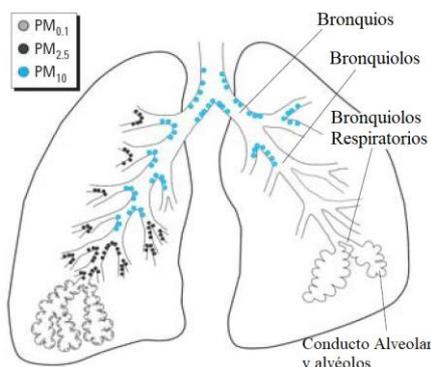
**Keywords:** convolutional neural networks, deep learning, pollutant particles, PM 10, prediction, CNN.

## 1. Introducción

La contaminación ambiental ha sido en la actualidad un tema de gran interés para un amplio sector de la población, debido a los estudios que se han realizado acerca de su incremento y los potenciales daños que pudiera ocasionar, tanto a la naturaleza como a la salud humana.

La materia particulada o partículas suspendidas (PM) son partículas contaminantes compuestas por distintos tipos de materia, tanto natural como provocada por el hombre [1]. Un subgrupo de las PM son las PM<sub>10</sub>, teniendo la característica de que son partículas con un diámetro menor a los 10 micrómetros. Su composición es muy variada, ya que una gran cantidad de materia está dentro de esas dimensiones. La composición de PM<sub>10</sub> ha sido estudiada en [1], donde se determina el porcentaje de cada elemento que compone las PM<sub>10</sub> siendo el carbono uno de los más grandes.

Por otra parte, el impacto de las altas concentraciones de PM<sub>10</sub> en la salud de las personas es elevado. Incluso concentraciones por debajo de las normas ambientales promedio tienen un impacto en la función respiratoria de los individuos [2]. Por dicha razón, el modelado de PM<sub>10</sub> es importante ya que es uno de los factores de mayor impacto a la salud de la población. En la figura 1 se observa un diagrama que permite hacer un comparativo entre los puntos donde se alojan comúnmente los distintos tipos de PM en el sistema respiratorio de las personas. Se pueden observar, aparte de PM<sub>10</sub>, las PM<sub>2.5</sub> que son partículas suspendidas con un diámetro menor a los 2.5 micrómetros y las menores a 0.1 micrómetros (PM<sub>0.1</sub>).



**Fig. 1.** Diagrama de puntos principales donde se concentran los subgrupos de PM en el sistema respiratorio [3].

Para el presente trabajo se hace uso de una base de datos pública de la cual se utilizan los datos de 7 variables meteorológicas que son medidas por 2 redes de monitoreo del Sistema de Monitoreo Atmosférico (SIMAT) de la ciudad de México, perteneciente a la Secretaría del Medio Ambiente (SEDEMA) del Gobierno Federal. Las redes de monitoreo consultadas tienen el nombre de Red Automática de Monitoreo Atmosférico (RAMA) y Red de Meteorología y Radiación Solar (REDMET). Estas redes de monitoreo son administradas por el Centro de Información de la Calidad del Aire (CICA), el cual recibe datos por minuto de todas las redes de monitoreo, posteriormente realiza

un promedio por hora para generar el Índice Metropolitano de la Calidad del Aire (IMECA) que se hace público cada hora en la página electrónica de la SEDEMA [4].

Las bases de datos utilizadas fueron consultadas en la página oficial de la SEDEMA, mediante la REDMET y RAMA. El SIMAT cuenta con 45 estaciones de monitoreo distribuidas en distintas zonas de la Ciudad de México y la zona Metropolitana (Fig. 2).

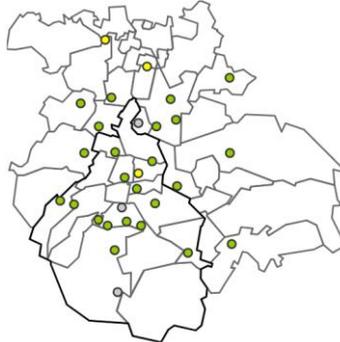


Fig. 2. Mapa de la localización de las redes de monitoreo del SIMAT [5].

Una vez que se obtuvieron los datos, se procedió a la implementación del algoritmo que permitiría el modelado de la concentración de PM10 y su variación, medido en unidades de microgramo por metro cúbico ( $\mu\text{g}/\text{m}^3$ ). En los últimos años las CNN han sido utilizadas para predicción de series de tiempo, logrando un buen desempeño; demostrando que tienen un buen potencial en este campo. Las CNN tienen la cualidad de encontrar patrones en grandes volúmenes de datos, al aplicar transformaciones al conjunto de datos.

## 2. Antecedentes

Se han implementado muchas técnicas para el modelado y predicción de PM10, algunas de ellas mediante la clasificación de PM10 en grupos dependiendo de su concentración como en [6] y [7], pero este acercamiento no da un valor exacto de predicción, sino un intervalo. Posteriormente en acercamientos más recientes se ha realizado una predicción de forma continua, lo cual entrega un resultado con una mayor certeza.

En [8] se implementa una red híbrida de CNN con Memoria a Corto y Largo Plazo (LSTM) para la predicción de ozono, logrando obtener resultados de precisión superiores a los previamente obtenidos mediante otras técnicas. Lo cual permite reconocer el potencial de las CNN para la predicción de series de tiempo.

Acercamientos previos que se han tenido al modelado y predicción de PM10, mediante algoritmos de optimización por enjambre, se pueden consultar en [9], donde se utiliza Optimización por Colonia de Hormigas para el modelado de PM10 y en [10] donde se utiliza Optimización por Enjambre de Partículas (PSO) en un Sistema de Inferencia Neuro-Difuso Adaptativo (ANFIS) para la predicción de dicho contaminante, logrando grandes mejoras por medio de PSO. Implementaciones de Máquina de Vectores de Soporte (SVM) para la predicción de Ozono y PM10 pueden consultarse en [11], donde se prueban distintos kernels para la predicción.

En [12] se hace uso de técnicas de clusterización para la selección de las mejores características para una arquitectura de red neuronal artificial (ANN).

Por último, en [13] se hace uso del Perceptrón Multicapa (MLP) para la predicción de concentraciones de PM10.

En el estado del arte, hasta el momento, no se ha reportado el uso de redes convolutivas para la predicción de ningún tipo de PM. El uso de una técnica distinta, no reportada previamente, será la principal aportación del trabajo de investigación.

### 3. Materiales y métodos

#### 3.1. Conjunto de datos

El conjunto de datos utilizado consta de la concentración de PM10 y 6 variables meteorológicas obtenidas de las redes de monitoreo. Algunas variables disponibles en la base de datos no fueron tomadas en cuenta debido a que tenían una gran cantidad de datos faltantes o solo se tenía registro reciente de ellas. Las variables elegidas se tomaron en cuenta debido a que son las que comúnmente se han utilizado, en trabajos previos relacionados, para la predicción de PM10 mediante otras técnicas y de igual manera porque son las que cuentan con una mayor cantidad de datos válidos dentro de la base de datos utilizada. En la Tabla 1 se pueden observar las variables seleccionadas junto con su métrica.

**Tabla 1.** Variables utilizadas para el conjunto de datos.

Variable	Métrica	Red de Monitoreo
PM10	Microgramo/metro cúbico ( $\mu\text{g}/\text{m}^3$ )	RAMA
Temperatura (TMP)	Grados Celsius ( $^{\circ}\text{C}$ )	REDMET
Dirección del Viento (WDR)	Grados Azimut	REDMET
Velocidad del Viento (WSP)	Metros/segundo (m/s)	REDMET
Humedad Relativa (RH)	Porcentaje (%)	REDMET
Radiación Solar Ultravioleta tipo A (UVA)	Mili Watt/centímetro cuadrado ( $\text{mW}/\text{cm}^2$ )	RAMA
Radiación Solar Ultravioleta tipo B (UVB)	Dosis mínima de eritema sobre hora (MED/h)	RAMA

Se utilizaron los datos por hora desde el año 2010 hasta 2018, por lo que el conjunto de observaciones es superior a los 166,000 por variable utilizada, dando un total aproximado de 1.1 millones de datos. Se creó una función en la que podía realizarse el procesamiento por mes y haciendo una comparación se determinó que al utilizar el mes de enero se podía crear un mejor modelo de predicción. Se experimentó con todos los meses y el conjunto de datos completo, pero finalmente se utilizaron los datos del mes de enero por lo que la cantidad de datos entre las 7 variables es aproximadamente de 100,000. De todas las estaciones de monitoreo disponibles se optó por utilizar la de Merced (MER), debido a que contaba con menor cantidad de datos faltantes en la mayor parte de los años utilizados. De igual manera en la estación MER se contaba con la

medición de las 7 variables utilizadas en los años seleccionados, cosa que en otras estaciones no se tenía disponibilidad.

De esta manera al contar con aproximadamente 100,000 datos en el conjunto al tomar el mes de enero, por variable contamos con poco más de 14,000 datos. Esto nos dice que tendremos aproximadamente 14,000 lotes para procesar en nuestras 3 arquitecturas. Al realizar el proceso de entrenamiento, validación y pruebas de 3 arquitecturas en repetidas iteraciones para la validación de los resultados deriva en una gran cantidad de tiempo de cómputo con esa cantidad de datos. Por lo que el uso del conjunto de datos completo tomaría 12 veces más en realizarse. Por esas razones se hace uso de una muestra mediante el mes de enero, resaltando que mediante experimentación fue el mes que generó un mejor resultado y por ende una mejor representación del conjunto total de datos.

### 3.2. Normalización de los datos

El conjunto de datos fue normalizado ya que durante la experimentación se determinó que presentaba mejores resultados de predicción al implementar la normalización.

Se presenta el estudio de diversos métodos de normalización en [14], en ese trabajo se concluye que la normalización por vector es la más eficiente para su trabajo de predicción. Sin embargo, por medio de la experimentación que determinó que para este trabajo existían otros métodos de normalización que generaban un mejor rendimiento.

Se utilizó la normalización z-score, dicha normalización consiste en restar la media del conjunto de datos a cada elemento y posteriormente dividir entre la desviación estándar como se muestra en la siguiente ecuación (ec. 1):

$$N_i = \frac{T_i - \mu_T}{\sigma_T}. \quad (1)$$

Una de las características de esta normalización es que el conjunto de datos resultante tiene una media de 0 y una desviación estándar de 1.

### 3.3. Creación de lotes de entrenamiento, validación y prueba

El conjunto de datos fue separado de la forma 80% de los datos para entrenamiento, 10% para validación y 10% prueba. Dado que se contó con un total de datos de 100,000 aproximadamente, se tuvieron 80,000 datos para entrenamiento, 10,000 para validación y 10,000 para pruebas.

El conjunto de datos fue organizado de tal manera que tuviera la forma necesaria de entrada para las arquitecturas de redes implementadas como se explica más adelante, esto debido a que el grupo de datos debía estar organizado en una matriz con el orden de los días previos de entrada, las 7 variables utilizadas y la cantidad de lotes de entrenamiento; por lo que se segmentó en una cantidad de entradas  $n$  (24 datos previos) para crear los lotes de entrenamiento, validación y prueba.

Debido a que se implementaron 3 arquitecturas, los datos fueron estructurados de la manera descrita a continuación. Las MLP requieren un vector bidimensional de la forma [Lotes, Entradas], las CNN de 1 dimensión (CNN1D) requieren una matriz 3D de la forma [Lotes, Entradas, Características] y las CNN de 2 dimensiones (CNN2D) requieren una matriz 4D de la forma [Lotes, Filas, Columnas, Canales].

### 3.4. Creación de las arquitecturas

Para tener un marco de comparación se utilizó un MLP (Fig. 3), similar al de [13], con una cantidad de neuronas de entradas dependiente de la cantidad  $n$  de datos de entrada, el cual fue determinado como 24, seguida de una capa oculta de 10 neuronas y una capa de salida con 1 neurona para la predicción de la hora siguiente. Se realizaron pruebas para determinar el mejor valor para la cantidad de neuronas en la capa oculta, resultando el valor de 10 como el más adecuado, a diferencia de los resultados que se obtuvieron en [13], donde se usaron dos modelos de MLP con 8 y 19 neuronas en la capa oculta respectivamente. El MLP se usa como comparación ya que es de las técnicas más recientes y con mayor uso en esta aplicación específica.

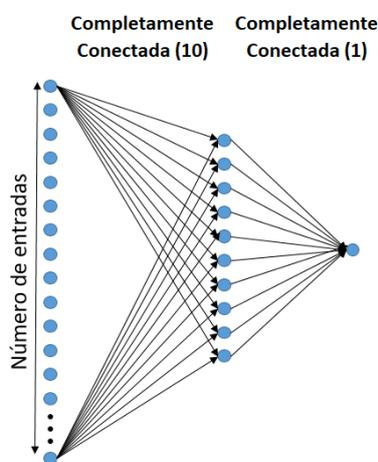


Fig. 3. Diagrama de MLP.

Se creó una CNN1D (Fig. 4) con una entrada  $[n, 7]$ , por las  $n$  entradas y 7 debido a que son 7 características para la predicción; esta capa fue seguida de una capa de aplanamiento que linealiza la salida de la CNN1D para que pueda ser procesada por la siguiente capa oculta de 10 neuronas, posteriormente se añadió una capa de deshabilitación de neuronas (Dropout) con una probabilidad de 15%, esto para evitar el sobreentrenamiento, finalmente una capa de salida de 1 neurona.

Finalmente se implementó una CNN2D (Fig. 5) con una entrada de  $[n, 7, 1]$ , las dos primeras dimensiones similares a la capa CNN1D y la última dimensión porque solo se utilizó un canal; posteriormente se insertó la capa de aplanamiento, una capa oculta de 10 neuronas, la capa de Dropout con 15% y una capa de salida de 1 neurona.

La principal diferencia entre las arquitecturas CNN1D y CNN2D es el kernel utilizado, ya que en la CNN1D el kernel es un vector de una fila por una cantidad  $n$  de columnas, por lo cual se realiza la convolución a cada fila de la matriz de entrada. Por otra parte, la CNN2D tiene un kernel de  $n$  filas por  $m$  columnas por lo que se realiza la convolución de manera bidireccional a la matriz de entrada; este último toma más datos para la convolución por lo que podría decirse que encuentra relaciones entre los datos en un mayor espacio de búsqueda.

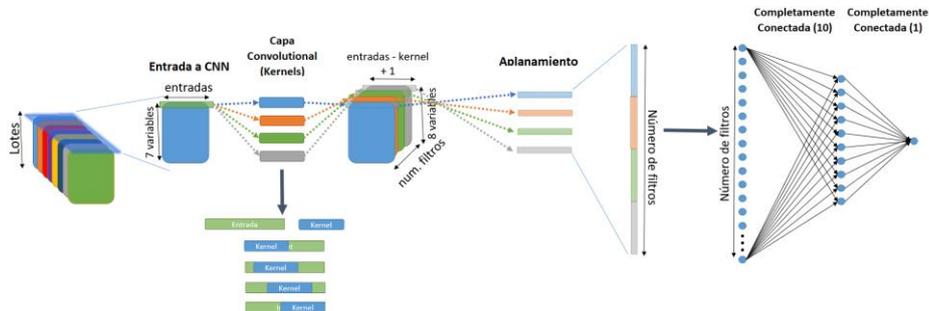


Fig. 4. Diagrama de CNN1D.

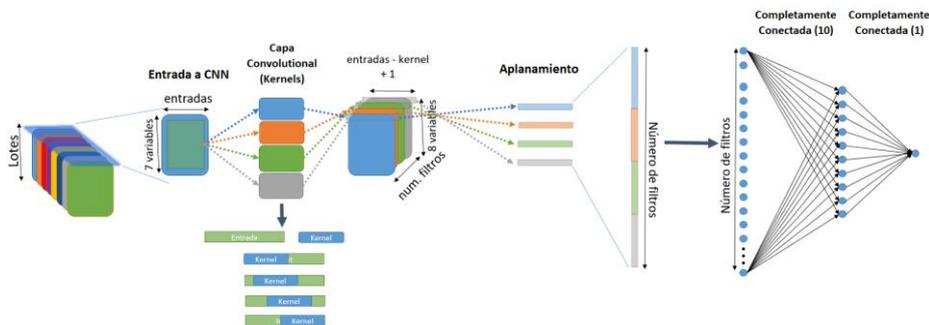


Fig. 5. Diagrama de la CNN2D.

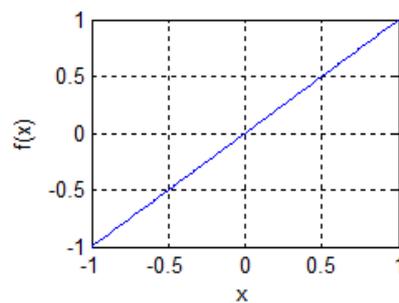


Fig. 6. Función de activación lineal.

Todas las capas de todas las arquitecturas tienen una función de activación lineal, como se muestra en la ecuación 2 y (Fig. 6), dicha función de activación se elige ya que los datos son normalizados y se requiere que las salidas puedan tener valores negativos. Otras funciones de activación tienden a llevar los valores de salida a 0 o limitan la salida a números positivos. Por otra parte, todas las arquitecturas usan el optimizador Adam; dicho optimizador ha demostrado tener un buen rendimiento en datos donde el gradiente es altamente variable [15]. Debido a que el conjunto de datos es altamente no lineal, mediante el uso de otro tipo de optimizadores como lo es el Gradiente de Descenso Estocástico (SGD), la búsqueda de un espacio de solución es más lento y no logra adecuarse a las grandes variaciones de los datos. Por su parte, Adam tiene un buen

desempeño en grandes conjuntos de datos y es apropiado para datos no estacionarios y con grandes variaciones en su gradiente [15]:

$$f(x) = x. \quad (2)$$

#### 4. Resultados

Se probó cada arquitectura con los mismos parámetros, éstos pueden observarse en la Tabla 2, por 10 iteraciones para obtener un promedio de los valores resultantes.

**Tabla 2.** Parámetros para las arquitecturas implementadas.

Parámetro	Método / Variable
Número de Entradas	50
Tamaño de Lote	100
Función de Activación	Lineal
Función de Costo	Media del Error Absoluto (MAE)
Optimizador	Adam

Adicionalmente, las capas de convolución se definieron con un tamaño de kernel igual al número de características, en el caso de CNN1D, y número de características por número de entradas para el caso de CNN2D. Ambas redes aplicaron 100 filtros a cada lote de entrada.

La función de costo, MAE, fue seleccionada debido a que fue la que presentó un mejor desempeño en las 3 arquitecturas. Ésta es calculada por medio de la ecuación (3), donde  $y$  es el valor real y  $\hat{y}$  es el valor predicho:

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j|. \quad (3)$$

Otra de las métricas utilizadas es la de la Raíz de la Media de los Errores al Cuadrado (RMSE), denotado por la ecuación (4), con la notación previamente definida:

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}. \quad (4)$$

La métrica de Media del Error Absoluto Porcentual da un valor que refleja la lejanía que tiene un punto predicho con respecto al valor real, una de las desventajas es que mientras mayores sean los valores del conjunto de datos, se puede presentar un mayor error. La ecuación (5) es utilizada para su cálculo:

$$MAPE = \frac{1}{n} \sum_{j=1}^n \left| \frac{y_j - \hat{y}_j}{y_j} \right|. \quad (5)$$

Adicionalmente se utiliza el Índice de Concordancia (IOA), obtenido mediante la ecuación (6). Se toma en cuenta la notación anterior y  $\mu_T$  corresponde a la media del conjunto de datos:

$$IOA = 1 - \frac{\sum_{j=1}^n (y_j - \hat{y}_j)^2}{\sum_{j=1}^n (|\hat{y}_j - \mu_T| + |y_j - \mu_T|)^2}. \quad (6)$$

En la figura 7 se puede observar una gráfica con la comparación entre el RMSE de cada arquitectura, se puede notar que el desempeño de ambas arquitecturas con capa de convolución otorgan un resultado ligeramente superior al MLP. Cabe mencionar que a menor valor de RMSE, menor error; por lo tanto, el modelo tiene una mayor precisión. Esto es notorio para todas las iteraciones con capa de convolución (tanto CNN1D como CNN2D), con excepción de la prueba número 7 (iteración) en donde CNN1D da un resultado de error ligeramente superior que MLP. Sin embargo, en general los resultados son mejores utilizando esta métrica de error con las capas de convolución en comparación con MLP.

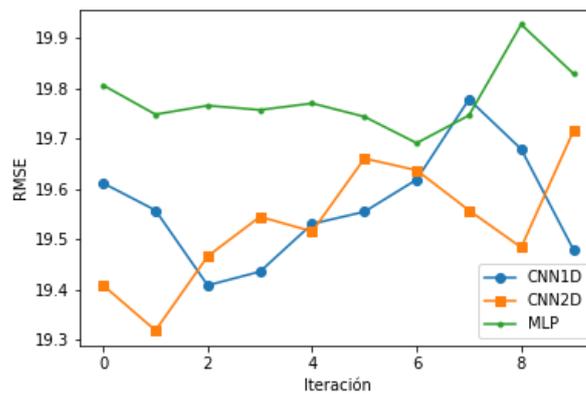
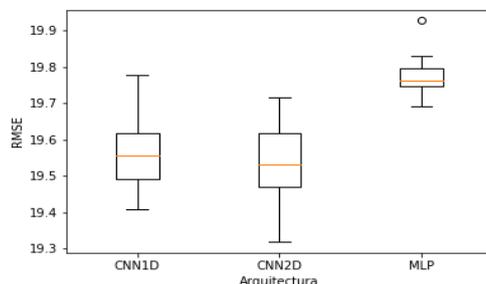


Fig. 7. Comparación de RMSE en 10 iteraciones con cada arquitectura.

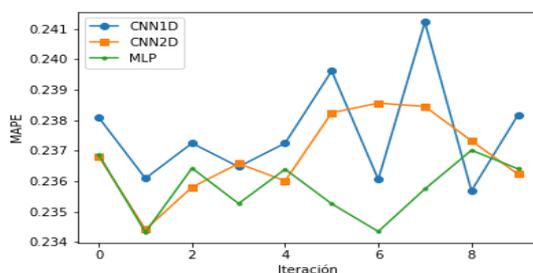
Como otro método de visualización en la figura 8 se muestra un diagrama de cajas con las mismas 10 iteraciones para el RMSE.

En la figura 8, es notorio que a pesar que la variabilidad es mayor para los resultados con arquitecturas de convolución, los resultados son en su gran mayoría inferiores, en cuestión de error, comparado con la misma métrica en MLP, lo que muestra una repetitividad en resultados con mayor precisión (por lo tanto, con un menor RMSE).

Por otra parte, el MAPE, resulta ser ligeramente menor para las mismas iteraciones, lo cual refleja la variabilidad de dicha métrica. En la figura 9 se muestra la comparación de los resultados de la experimentación. En suma, MAPE para esta aplicación, no es recomendable para identificar métricas de error por su alta variabilidad en las 3 arquitecturas.

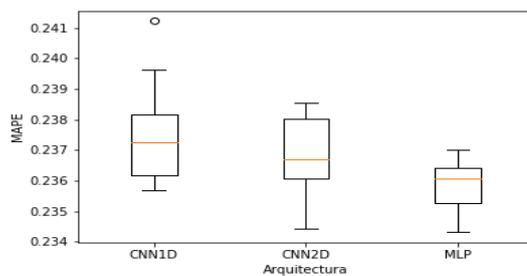


**Fig. 8.** Comparación de RMSE mediante diagrama de cajas.



**Fig. 9** Comparación de MAPE para las 3 arquitecturas.

El diagrama de cajas puede verse en la figura 10, en él puede notarse que en la CNN2D se alcanzan valores bajos similares a los de MLP, pero la media está por encima.



**Fig. 10.** Diagrama de cajas de MAPE.

Por último, se muestran los resultados del IOA, en la figura 11 mediante una gráfica con las 10 iteraciones y en la figura 12 mediante un diagrama de cajas. En el diagrama de cajas es más notorio que los resultados de MLP tienen mayor repetitividad, sin embargo, son más bajos.

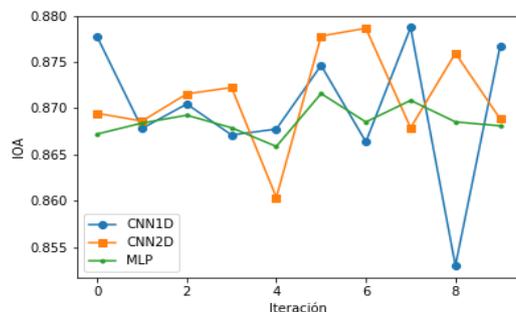


Fig. 11. Gráfica de comparación de IOA.

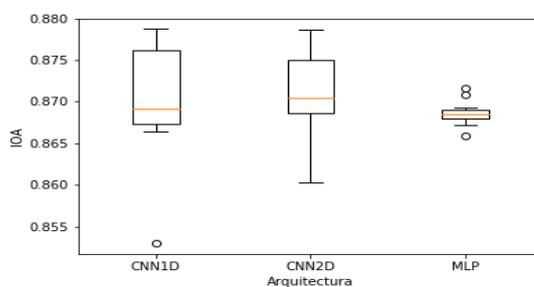


Fig. 12. Diagrama de cajas de IOA.

En la Tabla 3 se muestran los resultados en promedio del experimento realizado. Se puede observar que los resultados de las redes de convolución son ligeramente superiores a los de MLP en el caso de RMSE y de IOA, y menores en la métrica de MAPE, lo cual es un indicativo de que las redes convolutivas tanto de 1D como de 2D, tienen mejores resultados en prácticamente todos los casos con respecto a una red ampliamente utilizada en modelado de PM10 como lo es MLP.

Tabla 3. Promedio de Resultados para las 3 Arquitecturas.

Arquitectura	Métrica	Resultado
CNN1D	RMSE	19.5657
	MAPE	0.2376
	IOA	0.87
CNN2D	RMSE	19.5311
	MAPE	0.2368
	IOA	0.8711
MLP	RMSE	19.779
	IOA	0.8686

Haciendo un análisis de los resultados obtenidos puede observarse que el MLP, que en trabajos previos ha demostrado ser un mejor acercamiento a la predicción de PM10 que otros métodos, tiene un rendimiento ligeramente más bajo que ambas arquitecturas propuestas. La CNN1D obtuvo un mejor desempeño, con un RMSE ligeramente menor. Por su parte la CNN2D mejoró un poco más el RMSE. El RMSE es una de las métricas que mejor representa el desempeño junto con el IOA, ya que se basan en la diferencia del error predicho y el real por medio de las unidades que conciernen al modelo. Por su parte el MAPE es una unidad porcentual que tiene una gran variabilidad dependiendo de la magnitud de las mediciones, por lo que no es altamente confiable si se utiliza como única métrica de desempeño. Por medio de los resultados se puede decir que la CNN2D es la que otorga un modelo más confiable, aunque la diferencia es muy ligera comparativamente con CNN1D.

## 5. Conclusiones

Este trabajo presenta el uso de redes convolutivas para la predicción de PM10, dicha arquitectura no había sido utilizada hasta ahora con ese fin. De esta manera el trabajo de investigación tiene un impacto, ya que en el estado del arte no se reporta ningún trabajo que aborde el tema desarrollado. Las redes convolutivas tienen un buen desempeño para la predicción de series de tiempo, en el trabajo realizado han demostrado igualar y superar ligeramente una de las redes que en trabajos recientes habían demostrado el mejor desempeño para la predicción de PM10. Se hizo uso de un gran volumen de datos y de una gran cantidad de iteraciones para determinar que los resultados no son por la naturaleza estocástica de las redes neuronales; al hacer esto se asegura que los modelos propuestos son certeros y son robustos. Trabajos previos pueden dirigirse en el ajuste de los parámetros de las redes convolutivas para incrementar el desempeño, ya que existen varios parámetros que afectan el resultado; por mencionar algunos se tiene el tamaño de kernel, número de filtros, tamaño de lote, inicialización de bias y pesos, etc. Mediante el ajuste de los parámetros puede lograrse incrementar aún más la capacidad predictiva de las arquitecturas propuestas.

## Referencias

1. Chow, J.C., Watson, J.G., Edgerton, S.A., Vega, E.: Chemical composition of PM2.5 and PM10 in Mexico City during winter 1997. *Science of the Total Environment* 287(3), 177–201 (2002)
2. Pope Iii, C.A., Dockery, D.W., Spengler, J.D., Raizenne, M.E.: Respiratory Health and PM10 Pollution A Daily Time Series Analysis. *American Review of Respiratory Disease* 144, 668–674 (1991)
3. Cormier, S.A., Lomnicki, S., Backes, W., Dellinger, B.: Origin and health impacts of emissions of toxic by-products and fine particles from combustion and thermal treatment of hazardous wastes and materials. *Environmental Health Perspectives* 114(6), 810–817 (2006)
4. Calidad del Aire, <http://www.aire.cdmx.gob.mx/default.php?opc=%27aKBhnmM=%27>. Último acceso: 2019/03/29
5. Calidad del Aire Página Principal, <http://www.aire.cdmx.gob.mx/default.php>. Último acceso: 2019/03/29

6. Sun, W., Zhang, H., Palazoglu, A., Singh, A., Zhang, W., Liu, S.: Prediction of 24-hour-average PM 2.5 concentrations using a hidden Markov model with different emission distributions in Northern California. *Science of the Total Environment* 443, 93–103 (2013)
7. Dong, M., Yang, D., Kuang, Y., He, D., Erdal, S., Kenski, D.: PM 2.5 concentration prediction using hidden semi-Markov model-based times series data mining. *Expert Systems with Applications*, 36(5), 9046–9055 (2010)
8. Pak, U., Kim, C., Ryu, U., Sok, K., Pak, S.: A hybrid model based on convolutional neural networks and long short-term memory for ozone concentration prediction. *Air Quality, Atmosphere & Health* 11(8), 883–895 (2018)
9. Aceves-Fernández, M.A., Estrada, A.L., Pedraza-Ortega, J.C., Gorrostieta-Hurtado, E., Tovar-Arriaga, S.: Design and Implementation of Ant Colony Algorithms to Enhance Airborne Pollution Models. *International Journal of Environmental Science and Toxicology Research* 3(2), 22–28 (2015)
10. De León, B.O., Aceves-Fernandez, M.A., Fernández-Fraga, S.M., Ramos-Arreguín, J.M., Gorrostieta-Hurtado, E.: An improved particle swarm optimization (PSO): method to enhance modeling of airborne particulate matter (PM10). *Evolving Systems* 2019(0), 1–10 (2019)
11. Sotomayor-Olmedo, A., Aceves-Fernández, M.A., Gorrostieta-Hurtado, E., Pedraza-Ortega, C., Ramos-Arreguín, J.M., Vargas-Soto, J.E.: Forecast Urban Air Pollution in Mexico City by Using Support Vector Machines: A Kernel Performance Approach. *International Journal of Intelligence Science* 2013(3), 126–135 (2013)
12. Cortina-Januchs, M.G., Quintanilla-Dominguez, J., Vega-Corona, A., Andina, D.: Development of a model for forecasting of PM10 concentrations in Salamanca, Mexico. *Atmospheric Pollution Research* 6(4), 626–634 (2015)
13. Paschalidou, A.K., Karakitsios, S.: Forecasting hourly PM 10 concentration in Cyprus through artificial neural networks and multiple regression models: implications to local environmental management. *Environ Sci Pollut Res* 18, 316–327 (2011)
14. Panigrahi, S., Karali, Y., Behera, H.S.: Normalize Time Series and Forecast using Evolutionary Neural Network. *International Journal of Engineering Research & Technology* 2(9), 2518–2522 (2013)
15. Kingma, D.P., Ba, J.L.: ADAM: A Method for Stochastic Optimization. *ICLR* 1–15 (2015)