



Universidad Autónoma de Querétaro Facultad de Ingeniería

Modelado y Predicción de Partículas Contaminantes PM10 mediante Redes Profundas Convolucionales

Tesis

Que como parte de los requisitos para
Obtener el Grado de
Maestro en Ciencias en Inteligencia Artificial

Presenta

Ricardo Domínguez Guevara

Dirigido por:

Dr. Marco Antonio Aceves Fernández

Querétaro, Qro. a 22 de Enero de 2020



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Maestría en Ciencias en Inteligencia Artificial

Modelado y Predicción de Partículas Contaminantes PM10 mediante Redes
Profundas Convolucionales

Tesis

Que como parte de los requisitos para obtener el Grado de
Maestro en Ciencias en Inteligencia Artificial

Presenta

Ricardo Domínguez Guevara

Dirigido por:

Dr. Marco Antonio Aceves Fernández

Dr. Marco Antonio Aceves Fernández
Presidente

Dr. Efrén Gorrostieta Hurtado
Secretario

Dr. Juan Manuel Ramos Arreguín
Vocal

Dra. Carla María Quezada Angulo
Suplente

Dr. Jesús Carlos Pedraza Ortega
Suplente

Centro Universitario, Querétaro, Qro.
Fecha de aprobación por el Consejo Universitario (Febrero 2020)
México

Dedicatorias

Este trabajo está dedicado a Ángel, Estefanía, Hilda y Samuel, mi familia. Siempre han estado a mi lado brindándome su apoyo y me han dado las herramientas para enfrentarme a cualquier reto. Mi motivación y ánimo para superarme día a día, viene de ustedes.

Dirección General de Bibliotecas UAG

Agradecimientos

En primer lugar quiero agradecer a la Universidad Autónoma de Querétaro por darme la oportunidad de cursar mis estudios en el programa de Maestría, ya que me ha permitido tener el desarrollo académico y personal que buscaba.

Quiero agradecer también al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo brindado durante mis estudios, haciendo posible la conclusión de este trabajo de investigación.

El presente trabajo no hubiera sido posible sin el apoyo de todos mis profesores en la Universidad Autónoma de Querétaro, de cada uno aprendí cosas muy valiosas para mi desarrollo, tanto personal como académico. Especialmente, quiero agradecer al Dr. Marco Antonio Aceves Fernández por haber sido mi director de tesis. Siempre conté con su apoyo y su guía para desarrollar cada parte de este trabajo de investigación.

A mis compañeros y amigos, con los cuales viví este programa de Maestría. Este trabajo lleva el aporte de cada uno, ya que siempre conté con su apoyo.

Por último, agradezco a Ángel, Estefanía, Hilda y Samuel, mi familia. Este logro es también de ustedes, ya que, sin su incondicional apoyo no hubiera podido superar este reto en el que me han acompañado. Han estado a mi lado en cada momento, motivándome para superarme cada día.

Índice

Índice	5
Resumen.....	13
Palabras Clave	13
Abstract	14
Keywords.....	14
1. Introducción	15
2. Antecedentes	19
2.1. Hipótesis	28
2.2. Objetivos	28
2.2.1. Objetivo general	28
2.2.2. Objetivos específicos.....	28
3. Materiales y métodos.....	29
3.1. Conjunto de datos	29
3.2 Normalización.....	34
3.3. Perceptrón multicapa	34
3.4. Redes neuronales convolucionales	35
3.4.1. Red neural convolucional unidimensional	36
3.4.2. Red neuronal convolucional bidimensional.....	38
3.5. Hiperparámetros	39
3.5.1. Función de Activación.....	39

3.5.2. Optimizador	43
3.5.3. Tamaño del kernel	44
3.6. Métodos de Ensamble	45
3.7. Métricas de cálculo del error del modelo.....	46
3.7.1. Error cuadrático medio (RMSE).....	47
3.7.2. Error de porcentaje absoluto medio (MAPE)	47
3.7.3. Índice de acuerdo (IOA)	48
3.7.4. Coeficiente de determinación (R2).....	48
4. Resultados.....	50
5. Discusión de Resultados.....	69
6. Conclusiones y Trabajo Futuro	73
Referencias.....	74
Anexo I	80
Anexo II	94

Índice de Figuras

Figura 1.1.- Diagrama de los pulmones y localización de zonas donde quedan atrapadas las PM10, PM2.5 y PM0.1. (Cormier et al., 2006)	16
Figura 3.1. Gráfico de una muestra del conjunto de datos que muestra la concentración horaria de PM10 desde enero de 2000 hasta marzo de 2000. Se pueden ver las altas variaciones en la concentración en ese período de tiempo.	31
Figura 3.2. Comportamiento de las concentraciones de PM10 por año en el mes de enero desde 2000 hasta 2018. Los diagramas de caja nos permiten ver las diferencias en la mediana, los cuartiles y los valores atípicos.....	32
Figura 3.3. Comportamiento de las concentraciones de PM10 por año en el mes de agosto desde 2000 hasta 2018. Los diagramas de caja nos permiten ver las diferencias en la mediana, los cuartiles y los valores atípicos.....	32
Figura 3.4. Comportamiento de las concentraciones de PM10 por año en el mes de diciembre desde 2000 hasta 2018. Los diagramas de caja nos permiten ver las diferencias en la mediana, los cuartiles y los valores atípicos.	33
Figura 3.5. Arquitectura de Perceptrón Multicapa. [Creación Propia].....	35
Figura 3.6. Visualización del kernel unidimensional deslizándose a través de la entrada. [Creación Propia].....	36
Figura 3.7. Arquitectura de la red neuronal convolucional unidimensional (1DCNN). [Creación Propia].....	37
Figura 3.8. Visualización del kernel bidimensional deslizándose a través de la entrada. [Creación Propia].....	38
Figura 3.9. Arquitectura de una red neuronal convolucional bidimensional. El mismo proceso de la Figura 9 tiene lugar, luego cada salida de kernel es aplanada y luego apilada para entrar en las neuronas de entrada. [Creación Propia]	39
Figura 3.10. Función de activación lineal. Es una de las funciones de activación más simples, es rápido de calcular y no tiene restricciones en la salida.....	40

Figura 3.11. Función de activación ReLU. Esta función de activación también es rápida de calcular, pero permite una transformación de los datos de entrada.....	41
Figura 3.12. Función de activación sigmoide. Esta función de activación es muy adecuada para problemas de clasificación, específicamente para la clasificación binaria. Es más lento de calcular en comparación con las funciones ReLU y de activación lineal.	42
Figura 3.13. Función de activación Softplus. Este también es más lento de calcular que el ReLU y la función de activación lineal, pero es muy adecuado para aplicaciones de regresión y no tiene un cambio drástico en los valores de salida como el ReLU.	43
Figura 3.14. Diagrama del conjunto de Bagging.	46
Figura 4.1. Distribución de errores entre predicciones y datos reales. En el eje X se presenta el error real de todo el conjunto de pruebas y su distribución.	67
Figura 4.2. Distribución de errores en todas las arquitecturas. PEJ: Ensamble final con datos de Enero. FEJ: Ensamble Final con todo el conjunto de datos.....	68
Figura 4.3. Diagrama de los componentes del modelo final para la predicción de PM10. [Creación propia].....	71

Índice de Tablas

Tabla 2.1.- Trabajos previos relacionados a la modelación y predicción de partículas ambientales.....	24
Tabla 2.1 (cont)- Trabajos previos relacionados a la modelación y predicción de partículas ambientales.....	25
Tabla 2.1 (cont)- Trabajos previos relacionados a la modelación y predicción de partículas ambientales.....	26
Tabla 2.2.- Aplicaciones comunes de CNN y CNN para predicción.....	27
Tabla 3.1. Variables utilizadas para la predicción de PM10.....	30
Tabla 4.1. Resultados de arquitecturas MLP y conjunto completo.....	50
Tabla 4.1 (cont). Resultados de arquitecturas MLP y conjunto completo.....	51
Tabla 4.2. Modelos que forman parte del ensamble de arquitecturas MLP y resultado del ensamble.....	52
Tabla 4.3. Resultados de 1DCNN con arquitecturas de una capa convolucional y conjunto completo.....	53
Tabla 4.4. Ensamble final para 1DCNN con arquitecturas de una capa convolucional.....	54
Tabla 4.5. Resultados de 1DCNN con dos arquitecturas de capas convolucionales y conjunto completo.....	55
Tabla 4.5 (cont). Resultados de 1DCNN con dos arquitecturas de capas convolucionales y conjunto completo.....	56
Tabla 4.6. Ensamble final para 1DCNN con dos arquitecturas de capas convolucionales.....	57
Tabla 4.7. Resultados de las arquitecturas 2DCNN y ensamble completo.....	58
Tabla 4.7 (cont). Resultados de las arquitecturas 2DCNN y ensamble completo.....	59
Tabla 4.7 (cont). Resultados de las arquitecturas 2DCNN y ensamble completo.....	60
Tabla 4.7 (cont). Resultados de las arquitecturas 2DCNN y ensamble completo.....	61

Tabla 4.8. Ensamble final para arquitecturas 2DCNN.	62
Tabla 4.9. Resultados de las cuatro arquitecturas de ensamble resultantes y ensamble completo.....	62
Tabla 4.9 (cont). Resultados de las cuatro arquitecturas de ensamble resultantes y ensamble completo.	63
Tabla 4.10. Ensamble final para las cuatro arquitecturas de conjuntos resultantes.	64
Tabla 4.11. Resultados de las arquitecturas finales y el conjunto.	65
Tabla 4.12. Modelo final de ensamble.....	66

Dirección General de Bibliotecas UAQ

Abreviaturas y siglas

AF – Activation Function – Función de activación.

AI – Artificial Intelligence / Inteligencia Artificial.

ANN – Artificial Neural Network / Redes Neuronales Convolucionales.

BEM – Bagging Ensemble Method / Método de Ensamble Bagging.

BFOA – Bacteria Foraging / Búsqueda de Bacterias

BP – Back Propagation / Retropropagación.

CNN – Convolutional Neural Network / Red Neuronal Convolutiva.

DCNN – Deep Convolutional Neural Network / Red Neuronal Convolutiva Profunda.

DL – Deep Learning / Aprendizaje Profundo.

GA – Genetic Algorithms / Algoritmos Genéticos

GSN – Generative Stochastic Network / Red Estocástica Generativa.

KS – Kernel Size / Tamaño de Kernel.

PM0.1 – Particulate Matter with diameter less than 1 micrometers / Partículas suspendidas con diámetro menor a 1 micrometros.

PM10 – Particulate Matter with diameter less than 10 micrometers / Partículas suspendidas con diámetro menor a 10 micrometros.

PM2.5 – Particulate Matter with diameter less than 2.5 micrometers / Partículas suspendidas con diámetro menor a 2.5 micrometros.

MLP – Multilayer Perceptron / Perceptrón Multicapa.

RBF – Radial Basis Function / Función de Base Radial.

ReLU – Rectified Linear Unit / Unidad de Rectificación Lineal.

RGSN – Recurrent Generative Stochastic Network / Red Estocástica Generativa Recurrente.

RH – Relative Humidity / Humedad Relativa.

RQ – Recurrence Quantification / Cuantificación de recurrencia

SEDEMA – Secretaría del Medio Ambiente.

SEN – Sequential Encoding Network / Redes de Codificación de Secuencias.

SVM – Support Vector Machine / Máquina de vector soporte

TGSN – Temporal Generative Stochastic Network / Red Estocástica Generativa Temporal.

TMP – Temperature / Temperatura.

UVA – Ultra Violet Radiation Type A / Radiación Ultravioleta Tipo A.

UVB – Ultra Violet Radiation Type B / Radiación Ultravioleta Tipo B.

WDR – Wind Direction / Dirección del Viento

WSP – Wind Speed / Velocidad del Viento

1DCNN – One-dimensional Convolutional Neural Network / Red Neuronal Convolutiva unidimensional.

2DCNN – Two-dimensional Convolutional Neural Network / Red Neuronal Convolutiva bidimensional.

Resumen

Las partículas con un diámetro inferior a 10 micrómetros (PM10) son hoy en día un tema de estudio importante, principalmente por el aumento de su concentración y su impacto en el medio ambiente y la salud pública. Este trabajo de investigación resume el uso de redes neuronales convolucionales (CNN) para pronosticar las concentraciones de PM10 basadas en variables atmosféricas. En este estudio en particular, se exploró el uso de redes neuronales convolucionales profundas (tanto 1D como 2D) para probar la viabilidad de estas técnicas en las tareas de predicción. Además, en este trabajo de investigación se utiliza un método de ensamble llamado *bagging* (BEM) para mejorar la precisión del modelo de predicción. Por último, se utiliza una técnica de predicción de PM10 bien conocida, llamada perceptrón multicapa (MLP), como comparación para mostrar la viabilidad, precisión y robustez del modelo propuesto. Como parte de los resultados de este trabajo, se encontró que las CNNs superan a las MLP, especialmente cuando se ejecutan usando modelos de ensamble.

Palabras Clave

PM10, Redes Neuronales Convolucionales, Predicción, CNN, Ensamble
Bagging

Abstract

Particulate matter with a diameter less than 10 micrometers (PM10) is today an important subject of study, mainly because of its concentration increasing and its impact on environment and public health. This research project summarizes the usage of convolutional neural networks (CNN) to forecast PM10 concentrations based on atmospheric variables. In this particular case-study, the use of deep convolutional neural networks (both 1D and 2D) was explored to probe the feasibility of these techniques in prediction tasks. Furthermore, in this research project an ensemble method called bagging (BEM) is used to improve upon the accuracy of the prediction model. Lastly, a well-known technique for PM10 forecasting, called multilayer perceptron (MLP) is used as a comparison to show the feasibility, accuracy and robustness of the proposed model. As part of the results of this project, it was found that the CNNs outperforms MLP, especially when they are executed using ensemble models.

Keywords

PM10, Convolutional Neural Networks, Forecast, CNN, Bagging Ensemble

1. Introducción

La contaminación ambiental es un tema de gran interés en la actualidad, debido a que su concentración ha ido en aumento desde hace algunos años y esto ha traído como consecuencia un creciente número de enfermedades debidas a la exposición continua de las personas a este ambiente dañino.

Las partículas PM10, siglas en inglés de *particulate matter*, son lo que se conoce como materia particulada o, más comúnmente, materia suspendida en el aire; una característica de dichas partículas es que cuentan con un diámetro menor a los 10 micrómetros. Las PM10 están compuestas por distintos elementos, en (Chow et al., 2002) se hace un análisis de las principales elementos encontrados según su concentración.

La importancia que tienen las partículas PM10, y por lo cual es elegida como nuestro tema de estudio, es que han sido relacionadas con el desarrollo de diversos tipos de enfermedades, principalmente del sistema respiratorio al quedar atrapadas en distintos puntos, como se puede ver en la Figura 1.1.

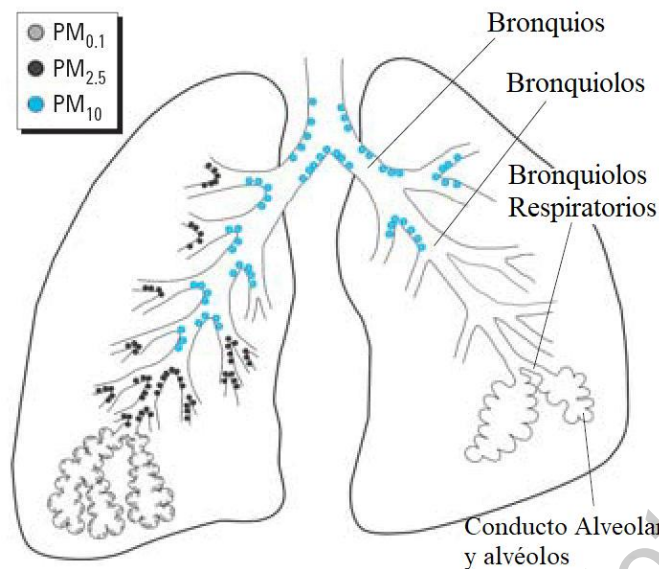


Figura 1.1.- Diagrama de los pulmones y localización de zonas donde quedan atrapadas las PM10, PM2.5 y PM0.1. (Cormier et al., 2006)

Las PM10 tienen la peculiar característica de que cuentan con el tamaño que les permite entrar, por medio de la respiración, al cuerpo de las personas. Una vez que entran al cuerpo pueden llegar a quedar atrapadas en los pulmones y con el paso del tiempo bloquear los canales por los cuales el aire entra, lo que puede ocasionar graves problemas respiratorios. Una vez que las PM10 han bloqueado los canales de entrada, se comienzan a desarrollar patologías que afectan la calidad de vida de la población.

En (Pope lli et al., 1991) se realiza un análisis que da como resultado una alta correlación negativa entre la concentración de PM10 y la disminución de funciones respiratorias en un grupo de personas. De la misma manera en (Khaniabadi et al., 2017) se estudian los efectos de PM10, NO₂ y O₃ en la salud de las personas mediante un estudio realizado en la ciudad de Kermanshah, Iran. En dicho trabajo, se identificó que el riesgo de mortalidad por problemas cardiovasculares aumentaba principalmente por la exposición a PM10, con 188 muertes; en comparación, la exposición a NO₂ y O₃ causó 33 y 83 muertes respectivamente.

Por lo anterior, el tener con anticipación un modelo que permita conocer cómo se desarrolla la concentración de partículas PM10, puede ser benéfico para la población. Teniendo la posibilidad de predecir con antelación los momentos en los que puede presentarse un aumento en la concentración de las partículas que genere un riesgo para las personas.

En la Ciudad de México, estas partículas han ido aumentando en grandes cantidades en los últimos años y eso es un tema de salud pública que hay que tener en cuenta. Es por ello que se colocaron estaciones de monitoreo en diferentes zonas de la ciudad, registrando las cantidades de estas partículas y otras relacionadas con ellas.

Los datos relacionados con estas partículas son altamente no lineales, es por eso que el modelado y predicción del comportamiento de PM10 puede obtenerse mediante el uso de técnicas inteligencia artificial. Se seleccionaron las redes neuronales convolucionales, debido a su capacidad de encontrar patrones en grandes cantidades de datos (Simard et al., 2003), para este trabajo se hace uso de un conjunto de aproximadamente 1.2 millones de datos. Debido a la naturaleza de los datos, altamente no lineal, se implementará una red multicapa, lo cual podría permitir el análisis de los datos en dimensiones mayores para poder obtener un modelo preciso.

El uso de técnicas de aprendizaje profundo; como lo son las redes neuronales convolucionales, se está comenzando a explotar, debido a que el desarrollo de hardware nos permite en la actualidad tener una mayor capacidad de procesamiento. Aprovechando este beneficio, el uso de grandes volúmenes de datos para su análisis y tratamiento nos otorga la posibilidad de crear modelos más precisos.

Un trabajo relacionado con la representación de datos en secuencias complejas por medio de modelos de aprendizaje automático puede revisarse en el trabajo de Beissinger (2018). En este estudio se hace uso de Redes Estocásticas Generativas (GSN), GSN Temporal (TGSN), GSN Recurrente (RGSN) y Redes de Codificación de Secuencias (SEN).

Por otra parte, se han realizado trabajos en los que las CNN han dado resultados favorables en tareas enfocadas a la predicción y el modelado. Kalchbrenner (2014) se usa una CNN para el modelado de enunciados, obteniendo un buen resultado en la relación entre palabras, logrando buenos resultados en clasificación respecto a sentimientos y preguntas. En el trabajo de Liu (2015) se realiza una predicción de la posición de fijación de la vista mediante diversos factores visuales.

Debido a esto, el uso de aprendizaje profundo con redes neuronales convolucionales es una gran herramienta para modelar el comportamiento altamente no lineal de las partículas PM10. Esto otorga la posibilidad de predecir con cierta probabilidad de acierto, los momentos en los que la concentración de dicho contaminante en el ambiente sea tan alta que pueda generar un riesgo en la población.

2. Antecedentes

El aprendizaje es un proceso muy complejo, depende de una gran cantidad de factores para que pueda suceder. En el ser humano se ha desarrollado de una manera en la que logramos hacer uso de nuestras capacidades cognitivas para distintos problemas. Sin embargo, es algo tan cotidiano para nosotros que no nos damos cuenta de la complejidad que este proceso tiene.

Desde la aparición de la computadora, se han tenido intentos por emular esta habilidad en estos sistemas. Hoy en día el desarrollo ha crecido en este aspecto, pero sin ninguna duda este problema es de los más desafiantes y retadores; la inteligencia artificial es el área que se encarga de estudiar los acercamientos para lograr este fin (Michalski et al., 1983).

Dentro del área de estudio de inteligencia artificial se encuentra una rama conocida como aprendizaje de máquinas, la cual se encarga del estudio y modelación computacional de los procesos de aprendizaje en sus distintas facetas. El campo de aprendizaje de máquinas está enfocado en tres áreas de investigación principales: la primera es el estudio orientado a la tarea, conocido como acercamiento ingenieril, en el cual se desarrollan sistemas de aprendizaje para mejorar el rendimiento en tareas específicas. El segundo es la simulación cognitiva, el cual se dedica a la investigación y simulación computacional de procesos humanos de aprendizaje. El tercero es el análisis teórico, el cual se enfoca en la exploración teórica en el espacio de métodos de aprendizaje posibles, independientemente de la aplicación que pueda tener (Michalski et al., 1983).

Una de las herramientas del aprendizaje de máquinas es la red neuronal artificial, cuyas siglas por su nombre en inglés son ANN, la cual consiste básicamente de un conjunto de unidades de procesamiento, llamadas neuronas, interconectadas mediante lazos dirigidos y ponderados. Desde su introducción por

McCulloch en 1943, han tenido un gran desarrollo. Una de sus mayores mejoras fue en 1986 cuando Rumelhart, Hinton y Williams desarrollaron el algoritmo de retropropagación (BP), el cuál es hasta la fecha uno de los algoritmos de aprendizaje más utilizados (Asson y Wang, 1990).

En 1979 se introdujo el Neocognitron, predecesor de las redes neuronales convolucionales (CNN), el cual contaba con una unidad convolucional. Diez años después, en 1989, se aplicó el algoritmo de BP a una red similar al Neocognitron. Se incluyeron capas convolucionales, la operación max-pooling, entre otras características, logrando la arquitectura fundamental de las CNN (Schmidhuber, 2015). Dichas CNNs han tenido desde su inicio un desempeño alto en tareas de aprendizaje visual, siendo aún uno de los referentes para este tipo de tareas.

Las redes convolucionales cuentan con ciertas características fundamentales, entre las partes principales que componen a una CNN están las capas convolucionales, la capa de max-pooling y la capa de unidad lineal rectificadas (ReLU). Jia (2014) hace uso de una arquitectura que cuenta con dichas características e incluye una capa de productos internos.

Las redes convolucionales son parte de lo que se conoce como aprendizaje profundo, el cual se caracteriza por ser una red neuronal con múltiples capas ocultas en su arquitectura. No existe una definición exacta del número de capas que convierten una red neuronal en red profunda, pero en términos generales se entiende que el aprendizaje profundo requiere de, al menos, 2 capas ocultas para aplicar transformaciones no lineales a los datos de su entrada.

Por lo anterior se considera adecuado el uso de una red neuronal convolucional profunda (DCNN) para el análisis de los datos PM10 ya que estos tienen un comportamiento altamente no lineal; la DCNN es capaz de realizar transformaciones

a los datos para encontrar patrones en su comportamiento y ajustar los pesos para conseguir un modelo confiable para los datos de entrada.

Acercamientos previos similares al problema que nos atañe, referentes al modelado de PM10, incluyen el uso de métodos de regresión y redes neuronales artificiales para la predicción de PM2.5 (similar a PM10 pero en dimensiones menores a 2.5 micrometros). Se realiza una comparación en el trabajo de Ordieres (2005), donde se registra el desempeño entre una red neuronal de tipo perceptrón multicapa, una red de función de base radial y una red de tipo perceptrón multicapa cuadrático; aquí encontraron que el mejor desempeño lo tiene la red de función de base radial, para esta aplicación. De igual manera se ha tenido un acercamiento a este problema mediante series de tiempo y regresión; Sun (2013) analiza el desempeño de un modelo oculto de Markov para la predicción de PM2.5 y en Dong (2010) analiza el desempeño de un modelo oculto semi-Markov para el mismo tipo de partículas; se menciona que el modelo ha mostrado predicciones acertadas, hasta 24 horas en antelación, ante casos de concentraciones extremas de PM2.5.

Dentro de las investigaciones más recientes, Sotomayor-Olmedo (2013) hace uso de support vector machines (SVM) y distintas funciones de kernel para predicción de contaminación del aire. En el trabajo de Cabrera-Hernandez (2019) se utiliza una novedosa técnica de aprendizaje, basado en sistemas biológicos, llamada Bacteria Foraging (BFOA) para la optimización de un algoritmo Neuro-Fuzzy con la finalidad de mejorar el proceso de predicción de PM10. En la misma línea de conocimiento, se usa un algoritmo Ant Colony para optimizar un algoritmo Neuro-Fuzzy para mejorar la predicción de PM10 (Martinez-Zeron et al., 2014). Otro acercamiento es presentado por Aceves-Fernandez (2014), en su trabajo describe el método para obtener características clave para el modelado de PM10 por medio de la Cuantificación de Recurrencia (RQ).

De igual manera, se han aplicado metodologías para optimizar los valores de entrada para la predicción de PM10 por medio de redes neuronales, Antanasijević (2013) hace uso de algoritmos genéticos (GA) con este fin. Algoritmos de agrupamiento por medio de aprendizaje no supervisado se han aplicado en el trabajo de Elangasinghe (2014), donde se utiliza el algoritmo de k-means para segmentar datos de PM10 en áreas geográficas que son posteriormente modeladas por redes neuronales.

Como puede observarse, los acercamientos al modelado y predicción de PM10 son bastos, sin embargo aún quedan áreas que no han sido exploradas y existen técnicas que no han sido reportadas en la literatura.

Dentro de los enfoques modernos están las redes neuronales convolucionales (CNN), que son una parte del aprendizaje de máquinas, de igual manera están dentro de las técnicas de aprendizaje profundo que recientemente ha tomado fuerza por sus buenos resultados en aplicaciones diversas, incluyendo la predicción y modelado (Busia y Jaitly, 2016). En cuanto al aprendizaje profundo, se realizó un análisis de las mejores prácticas para el uso de CNN en el trabajo de Simard (2003), donde describen que uno de los factores que mayor peso tiene, para la mejora de resultados en las CNN, es el conjunto de datos. Mientras mayor sea el conjunto de datos para el entrenamiento de la red, mejor será su desempeño.

Las CNN han sido explotadas ampliamente en ciertas áreas, más específicamente en el reconocimiento de imágenes (He, 2015), clasificación (Krizhevsky y Hinton, 2012) y detección (Girshick et al., 2012).

También han sido usadas, en el área de visión por computadora, para la predicción de profundidad, normales a superficies y etiquetas semánticas (Eigen et al., 2015). Esto para dotar a un sistema de la capacidad para determinar el ambiente en el que está.

El uso de CNN, ha sido explorado recientemente en tareas de predicción. Busia y Jaitly (2016) detallan su uso en la predicción de estructuras secundarias de proteínas, proponiendo una arquitectura basada en CNN que está apoyada por otras técnicas para lograr una predicción aceptable.

Las investigaciones más recientes enfocadas a la predicción de PM10 comienzan a utilizar algunas técnicas de aprendizaje profundo; Athira (2018) hace uso de redes neuronales recurrentes, memoria a corto y largo plazo y unidades recurrentes cerradas. Se hace una comparación de estos tres métodos y se determina que las unidades recurrentes cerradas cuentan con una predicción más acertada. Proponen, al final del artículo, el uso de redes neuronales convolucionales para su comparación con el desempeño en su trabajo. Así puede observarse que los riesgos a la salud ocasionados por PM10 han existido desde hace algunos años, y siguen existiendo hasta la fecha.

Finalmente Pak (2018) trabaja con la predicción de concentraciones de ozono mediante un modelo híbrido de redes neuronales convolucionales y memoria a corto y largo plazo. En el cual utilizan datos históricos de 34 días previos para la predicción de las siguientes 8 horas de concentración de ozono, logrando resultados muy acertados.

En la Tabla 2.1 se puede ver un compendio de los trabajos previos que se han dado con relación a la predicción de partículas ambientales, así como las distintas técnicas que se han implementado.

Tabla 2.1.- Trabajos previos relacionados a la modelación y predicción de partículas ambientales.

Autores	Año	Método Utilizado	Aplicación	Tipo de Dato
Odieres, J. B. Vergara, E. P. Capuz, R. S. Salazar, R. E.	2005	Perceptrón Multicapa / Función de Base Radial / Perceptrón Multicapa Cuadrático	Predicción de partículas finas	PM2.5
Dong, M. Yang, D. Kuang, Y. He, D. Erdal, S. Kenski, D.	2010	Modelo oculto de Markov	Predicción de concentración de PM2.5	PM2.5
Dong, M. Yang, D. Kuang, Y. He, D. Erdal, S. Kenski, D.	2010	Modelo oculto semi-Markov	Predicción de concentración de PM2.5	PM2.5
Paschalidou, A. K. Karakitsios, S. Kleanthous, S. Kassomenos, P. A.	2011	Perceptrón Multicapa / Función de Base Radial	Predicción de la concentración de partículas contaminantes.	PM10
Sun, W. Zhang, H. Palazoglu, A. Singh, A. Zhang, W. Liu, S.	2013	Modelo oculto de Markov	Predicción de concentración de PM2.5	PM2.5

Tabla 2.1 (cont).- Trabajos previos relacionados a la modelación y predicción de partículas ambientales.

Autores	Año	Método Utilizado	Aplicación	Tipo de Dato
Sotomayor-Olmedo, A. Aceves-Fernandez, M. A. Gorrostieta-Hurtado, E. Pedraza-Ortega, J. M. Ramos-Arreguín, J. M. Soto-Vargas, J. E.	2013	Máquina de vector soporte / kernel	Predicción de contaminación urbana.	PM10
Martinez-Zeron, E. Aceves-Fernandez, M. A. Gorrostieta-Hurtado, E. Sotomayor-Olmedo, A. Ramos-Arreguín, J. M.	2014	Algoritmo neuro-difuzo / Ant Colony	Mejora de la predicción de contaminantes.	PM10
Aceves-Fernandez, M. A. Pedraza-Ortega, J. M. Sotomayor-Olmedo, Artemio. Ramos-Arreguín, J. M. Vargas-Soto, J. E. et al.	2014	Cuantificación de recurrencia	Análisis de características clave de comportamiento de PM10.	PM10
Athira, V. Geetha, P. Vinayakumar, R. Soman, K. P.	2018	Redes Neuronales Recurrentes	Predicción de calidad del aire	PM10

Tabla 2.1 (cont).- Trabajos previos relacionados a la modelación y predicción de partículas ambientales.

Autores	Año	Método Utilizado	Aplicación	Tipo de Dato
Pak, U. Kim, C. Ryu, U. Sok, K. Pak, S.	2018	Redes Neuronales Convolucionales / Memoria a corto y largo plazo	Predicción de concentración de ozono	Ozono
Cabrera-Hernandez, M. C. Aceves-Fernandez, M. A. Ramos-Arreguín, J. M. Vargas-Soto, J. E. Gorrostieta-Hurtado. E.	2019	Algoritmo neuro-difuso / Bacteria Foraging	Optimización del proceso de predicción de la calidad del aire.	PM10

En la Tabla 2.2 se muestran algunos trabajos referentes a las CNN tanto en aplicaciones comunes como en predicción. Es importante notar que las aplicaciones en predicción son recientes debido a que este ámbito, por medio de CNN, no ha sido explorado de manera muy amplia (Li, Ding, y Sun, 2018).

Tabla 2.2.- Aplicaciones comunes de CNN y CNN para predicción.

Autores	Año	Método Utilizado	Aplicación	Tipo de Dato
Krizhevsky, H. Geoffrey, E.	2012	Redes Neuronales Convolucionales Profundas	Clasificación de imágenes	Imágenes
Girshick, R. Donahue, J. Darrell, T. Malik, J. Barkeley, U. C.	2012	Redes Neuronales Convolucionales	Detección de objetos	Imágenes
He, K.	2015	Redes Residuales Profundas / Redes Neuronales Convolucionales	Reconocimiento de imágenes	Imágenes
Busia, A. Jaitly, N.	2016	Redes Neuronales Convolucionales	Predicción de estructuras proteicas	Proteínas
Li, X. Ding, Q. Sun, J. Q.	2018	Redes Neuronales Convolucionales Profundas	Pronóstico de vida útil restante	Parámetros de turbinas

Puede verse que hay un gran trabajo de investigación enfocado a PM10, sin embargo, en la literatura no se registra el uso de CNN para modelado y predicción de PM10 por medio de series de tiempo. Éste es el tema que se estudiará en este trabajo.

2.1. Hipótesis

El uso de una arquitectura basada en redes neuronales convolucionales creará un modelo robusto del comportamiento de las partículas PM10 y permitirá la predicción de variaciones en la concentración.

2.2. Objetivos

2.2.1. Objetivo general

Diseñar un modelo robusto por medio de una arquitectura basada en redes neuronales convolucionales, entrenada con los datos preprocesados con valores de PM10 obtenidos de estaciones de monitoreo en la Ciudad de México, que permitirá el modelado y predicción de la concentración de PM10 en ciertas regiones de la Ciudad de México.

2.2.2. Objetivos específicos

- Utilizar un método de preprocesamiento de los datos que permita sustituir los valores atípicos en el conjunto de datos por valores representativos.
- Definir un modelo robusto de predicción de PM10 por medio de redes neuronales convolucionales mediante revisión de la literatura y pruebas experimentales para contar con un modelo predictivo confiable.
- Diseñar una arquitectura basada en redes neuronales convolucionales que reciba de entrada los datos de factores ambientales y PM10 de un tiempo anterior, para tener como salida la predicción de PM10.

3. Materiales y métodos

3.1. Conjunto de datos

El conjunto de datos fue obtenido de una base de datos pública de una dependencia del gobierno mexicano llamada Secretaría de Medio Ambiente (SEDEMA) mediante su página electrónica (Secretaría del Medio Ambiente, 2019). Esta base de datos contiene mediciones de diferentes variables ambientales en la Ciudad de México. Este conjunto de datos contiene datos horarios para cada variable desde 1986 (Calidad del Aire, 2019).

De esa base de datos se seleccionaron 7 variables. El criterio para esa selección fue la cantidad de datos no válidos que contenía cada variable. Las variables elegidas se describen en la Tabla 3.1. Algunos de los valores reportados en el conjunto de datos fueron mostrados como -99 en los casos en que el sensor tenía un error de medición. En ese sentido, las variables seleccionadas para este trabajo fueron las que presentaron menos errores de medición en los años 2000 a 2018.

Tabla 3.1. Variables utilizadas para la predicción de PM10.

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 (mW/cm^2)	RAMA
Radiación Solar Ultravioleta tipo B (UVB)	Dosis mínima de eritema sobre hora (MED/h)	RAMA

Posteriormente, los datos fueron preprocesados para reemplazar los errores de medición con un valor más representativo. Se puede observar en la Figura 3.1 que el conjunto de datos presenta un comportamiento altamente no lineal, lo cual es una característica que debe tomarse en cuenta.

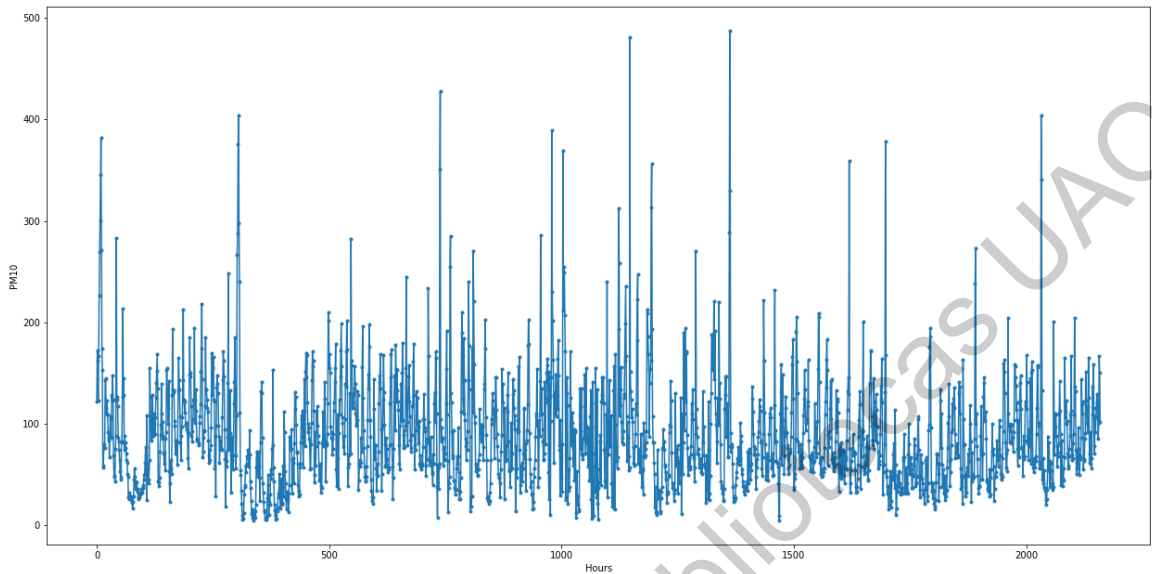


Figura 3.1. Gráfico de una muestra del conjunto de datos que muestra la concentración horaria de PM10 desde enero de 2000 hasta marzo de 2000. Se pueden ver las altas variaciones en la concentración en ese periodo de tiempo.

Con el fin de ahorrar tiempo de cálculo, se realizaron las pruebas con todos los modelos con datos de enero de cada año. Esta decisión se tomó en base a la experimentación, en la que se implementaron 13 modelos con los datos de cada mes y el conjunto completo de datos. Los resultados sugieren que enero es un buen subconjunto para tomar como muestra con fines de prueba. Con este paso los datos se reducen de aproximadamente 1.200.000 muestras a 100.000 muestras.

En la Figura 3.2 se muestra un diagrama de caja con la distribución de datos de enero desde 2000 hasta 2018 para observar el comportamiento de PM10 a través del tiempo.

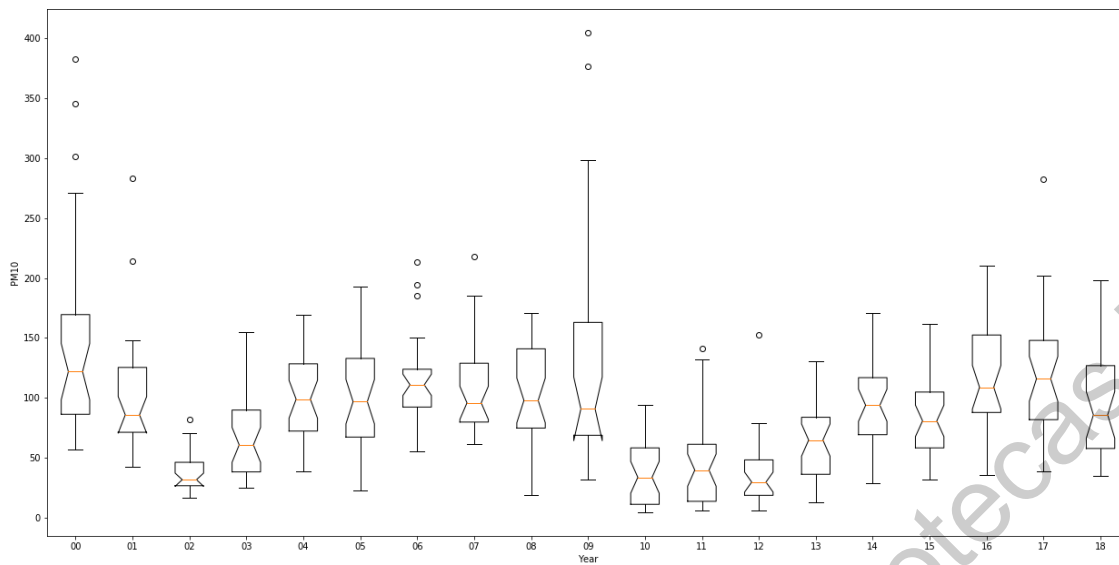


Figura 3.2. Comportamiento de las concentraciones de PM10 por año en el mes de enero desde 2000 hasta 2018. Los diagramas de caja nos permiten ver las diferencias en la mediana, los cuartiles y los valores atípicos.

En el mismo sentido, en la Figura 3.3 y en la Figura 3.4 se muestra el comportamiento de las PM10 en los meses de agosto y diciembre respectivamente.

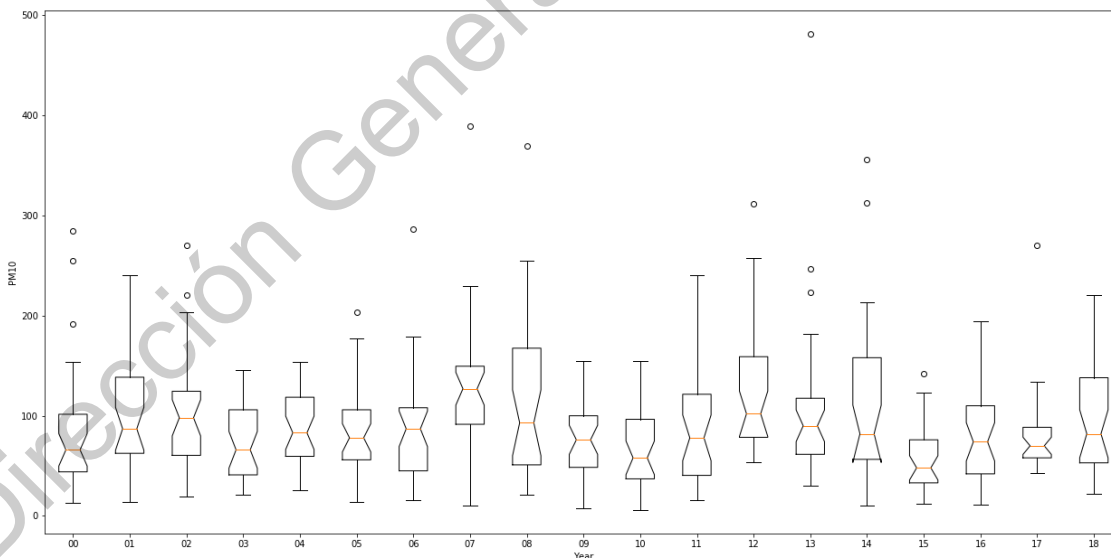


Figura 3.3. Comportamiento de las concentraciones de PM10 por año en el mes de agosto desde 2000 hasta 2018. Los diagramas de caja nos permiten ver las diferencias en la mediana, los cuartiles y los valores atípicos.

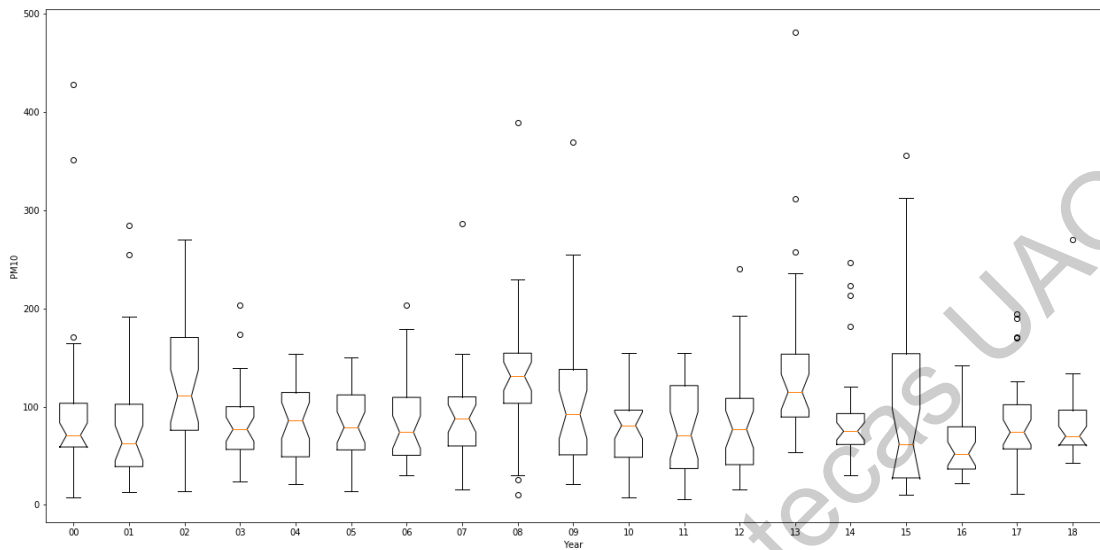


Figura 3.4. Comportamiento de las concentraciones de PM10 por año en el mes de diciembre desde 2000 hasta 2018. Los diagramas de caja nos permiten ver las diferencias en la mediana, los cuartiles y los valores atípicos.

De la Figura 3.2 a 3.4 se observa que las variaciones de concentración a través de cada mes y también a través de todos los años son consistentemente altas. Tomemos el caso de Enero, en el año 2000 a 2002 se tiene registrada una concentración con valor medio de PM10 de 125, 75 y 40, respectivamente. De 2000 a 2001, estos valores representan una disminución del 40%. De manera similar, del año 2000 al 2002, estos valores representan una disminución del 68%. Y por último, de 2001 a 2002, se tiene una disminución del 47%. Este comportamiento es dinámico en cada año con porcentajes de aumento y disminución variables.

Esta variabilidad implica un reto en el proceso predictivo; se requiere de un método de predicción que sea consistente en realizar buenas aproximaciones aún con la alta variabilidad del comportamiento. No se puede determinar un patrón evidente que permita aproximar la concentración en cada año. Resulta un problema mayor si se hace este análisis con base a los meses, días e incluso horas. El análisis aquí realizado se realiza en la predicción de concentración por hora, lo cual representa mayor variabilidad.

El aprendizaje profundo es uno de los acercamientos más novedosos para lograr esta clase de tareas complejas. Ya que permite entrenar una red neuronal artificial con grandes volúmenes de datos y detectar patrones que con otras técnicas es difícil de encontrar.

3.2 Normalización

En la mayoría de las aplicaciones se informa que los métodos de normalización son un factor importante que puede generar mejoras en la precisión de un modelo (Panigrahi et al., 2013), por esa razón se realizó una normalización de z-score en los datos de entrada.

La normalización del Z-score consiste en la sustracción de todos los valores originales (T_i) y la media del conjunto de datos (μ_T), luego dividida por su desviación estándar (σ_T). De esta manera se calcula el conjunto de datos normalizados (N_i). Esto se muestra en la ecuación 3.1.

$$N_i = \frac{T_i - \mu_T}{\sigma_T} \quad (3.1)$$

Una característica de esta normalización es que el conjunto de datos normalizados resultante tiene una media igual a 0 y una desviación estándar de 1.

3.3. Perceptrón multicapa

La arquitectura del perceptrón multicapa (MLP) es uno de los casos de estudio de esta investigación, porque es una de las arquitecturas más populares hoy en día para la previsión de PM10 (Paschalidou et al., 2011; Ordieres et al., 2015; Abderrahim et al. 2016; Cortina–Januchs et al. 2015; Kurt et al. 2008). MLP normalmente consiste en una capa de entrada, una o dos capas ocultas y una capa de salida. Todos ellos con neuronas totalmente conectadas entre capas consecutivas (Figura 3.5).

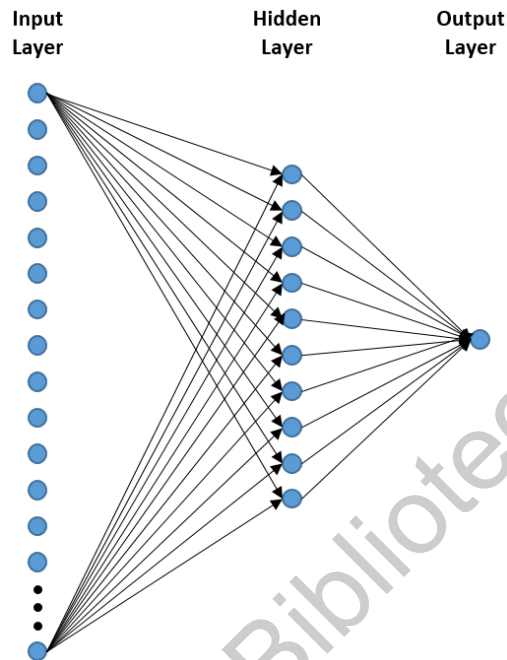


Figura 3.5. Arquitectura de Perceptrón Multicapa. [Creación Propia]

Esta es una representación simple de la MLP, la capa de entrada no tiene un número definido de neuronas porque puede depender del número deseado de horas previas en esta aplicación. La capa oculta se define con 10 neuronas y la capa de salida tiene una neurona porque es el número de horas que queremos predecir.

3.4. Redes neuronales convolucionales

La arquitectura propuesta para este trabajo es CNN, que fue seleccionada por su capacidad de extracción de características de los datos de entrada (Li et al., 2018). Esta característica presente en CNN se debe a la aplicación de un kernel a través de los datos de entrada. Este núcleo es básicamente una matriz de n filas y m columnas, donde n debe ser menor que las filas totales de los datos de entrada y m debe ser menor que las columnas totales de los datos de entrada.

En esta investigación, se seleccionaron dos tipos de CNN: CNN unidimensional (1DCNN) y CNN bidimensional (2DCNN). La principal diferencia de estas arquitecturas es el núcleo utilizado para realizar la convolución de los datos de entrada. En el 1DCNN el núcleo se desliza en una dimensión a través de los datos y en el caso del 2DCNN el núcleo se desliza en dos dimensiones a través de los datos. El rendimiento puede depender de la aplicación en la que se hace uso de la CNN, debido a esto ambos métodos son sometidos a experimentación para determinar el comportamiento de dichos métodos para este caso de estudio.

3.4.1. Red neural convolucional unidimensional

Este tipo de CNN consiste en una capa de convolución que aplica un filtro unidimensional a los datos de entrada (Figura 3.6).

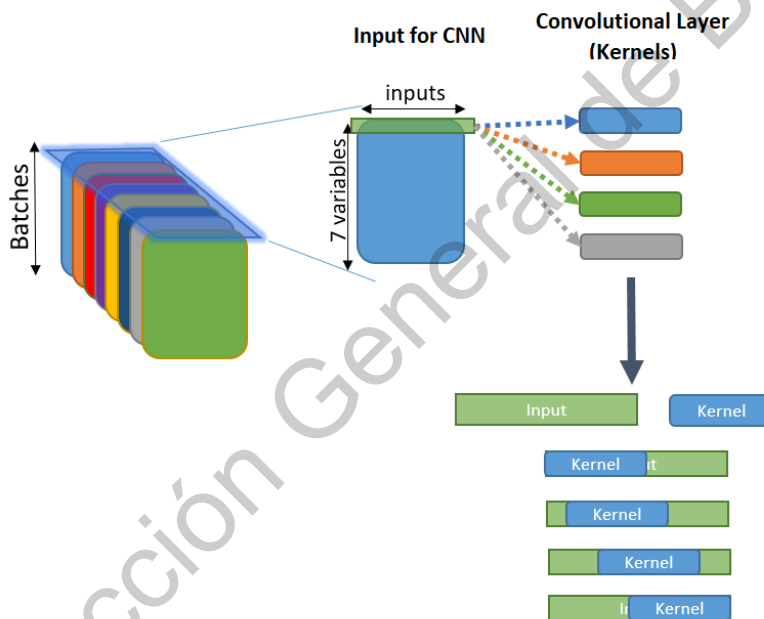


Figura 3.6. Visualización del kernel unidimensional deslizándose a través de la entrada. [Creación Propia]

Como se observa en la Figura 3.6, se visualiza el set de datos completo como entrada, se procesa en lotes de tamaño $7 \times n$ (donde n es el número de entradas).

La Figura 3.7 muestra los siguientes pasos, donde una capa aplanada da forma a la salida de los filtros de convolución para que sea un vector unidimensional. Luego, se agrega una capa completamente conectada con una caída del 15% para finalmente reducirla a una capa de salida con una neurona para obtener el pronóstico esperado.

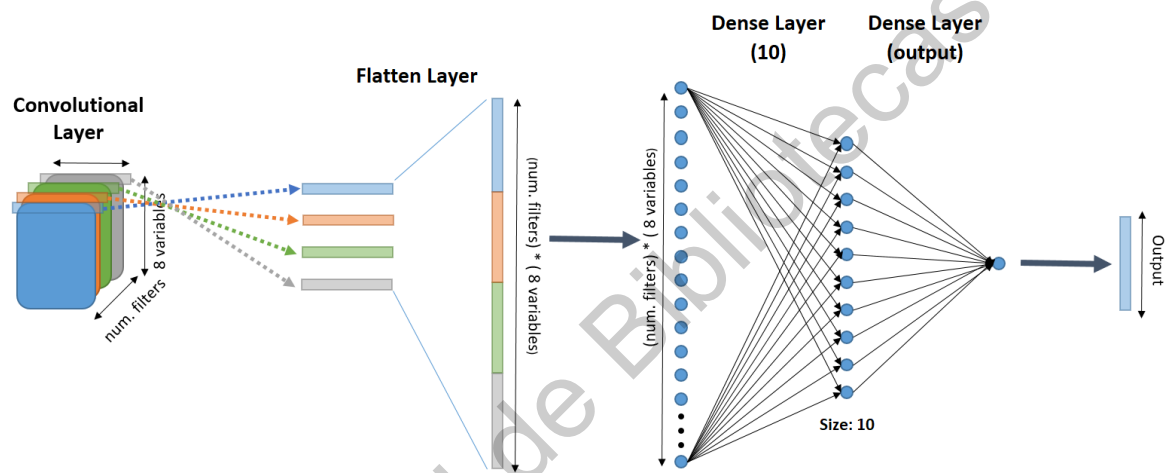


Figura 3.7. Arquitectura de la red neuronal convolucional unidimensional (1DCNN). [Creación Propia]

El proceso descrito en la Figura 3.7 tiene lugar en un inicio, después la salida de los *kernels* aplicados se aplanan para entrar en las neuronas de entrada.

Este tipo de CNN tiene una restricción de movimiento en el filtro de convolución, ya que sólo puede deslizar una dimensión a la vez, lo que restringe la extracción de características a una ventana más pequeña.

En este trabajo se utilizó 1DCNN con una capa convolucional y dos capas convolucionales. La segunda variante realiza otra ronda de transformaciones con nuevos núcleos a la salida de la primera capa de convolución.

3.4.2. Red neuronal convolucional bidimensional

2DCNNs puede ser un buen enfoque porque no tiene la restricción de movimiento sobre los datos de entrada como 1DCNN. La arquitectura básica es la misma que la de 1DCNN, la única diferencia es el tamaño del núcleo aplicado a los datos de entrada (Figura 3.8).

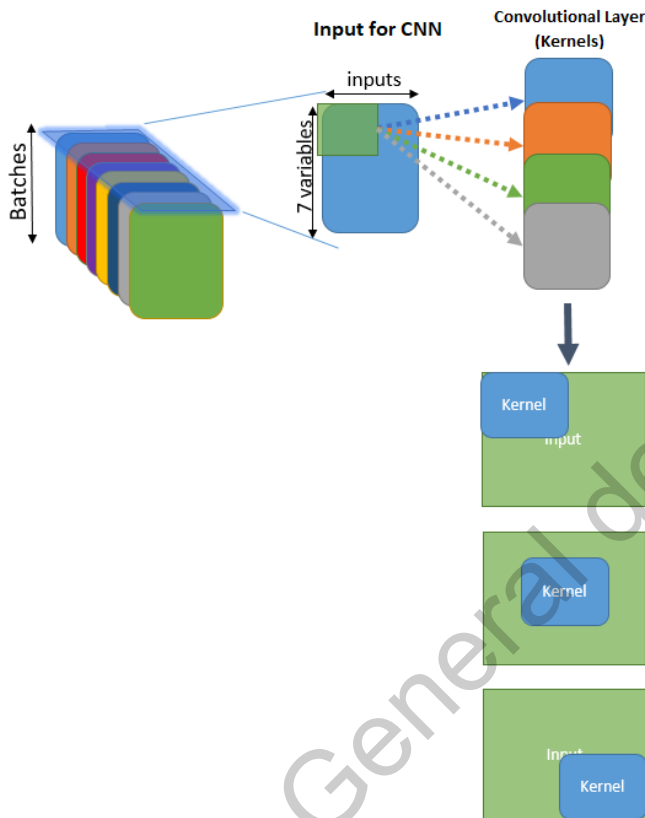


Figura 3.8. Visualización del kernel bidimensional deslizándose a través de la entrada.
[Creación Propia]

En la figura 3.8 se observa, que los núcleos son bidimensionales y los resultados son bidimensionales también.

En la Figura 3.9 se muestra una visualización completa de la arquitectura. Observe que el procedimiento seguido para esta arquitectura es el mismo que para 1DCNN. Con la diferencia del comportamiento y tamaño del núcleo.

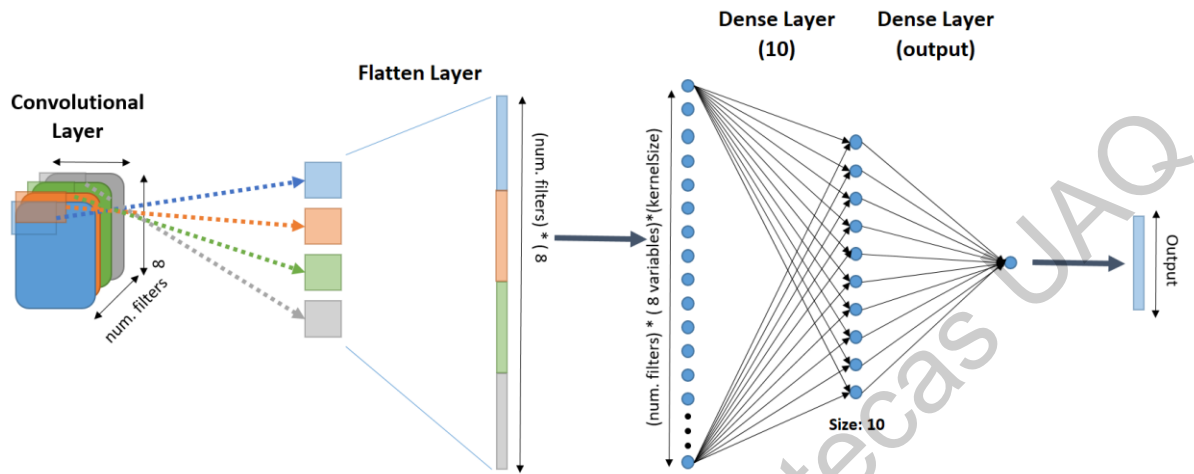


Figura 3.9. Arquitectura de una red neuronal convolucional bidimensional. El mismo proceso de la Figura 9 tiene lugar, luego cada salida de kernel es aplanada y luego apilada para entrar en las neuronas de entrada. [Creación Propia]

3.5. Hiperparámetros

Todos los parámetros internos que se pueden modificar en un modelo se denominan hiperparámetros, son cruciales para la precisión final del modelo y es ahí donde radica su importancia (Bergstra y Bengio, 2012). Estos hiperparámetros de 1DCNN y 2DCNN fueron ajustados para determinar en qué medida los modelos son capaces de predecir con precisión el comportamiento de los datos de PM10.

3.5.1. Función de Activación

Esta es una función que se realiza en cada neurona, su tarea es activar una neurona a diferentes intensidades dependiendo de la entrada recibida. Se probaron cuatro funciones de activación muy conocidas en las arquitecturas para determinar cuál es la mejor para esta aplicación, cuáles son: Unidad lineal, lineal rectificada (ReLU), Sigmoid y Softplus.

Algunas funciones de activación son más rápidas de calcular, y otras son mejores para algunas tareas debido a la forma en que procesan los datos de entrada. En (Shenouda, 2006) se hace una comparación de las funciones de activación en las que la función de activación lineal presenta un cálculo más rápido pero el peor rendimiento de todas las variaciones probadas en un problema de clasificación.

3.5.1.1. Lineal

La función de activación lineal permite básicamente que el valor de entrada se dispare como salida de la neurona, como se muestra en la ecuación 3.2 y en la Figura 3.10.

$$f(x) = x \quad (3.2)$$

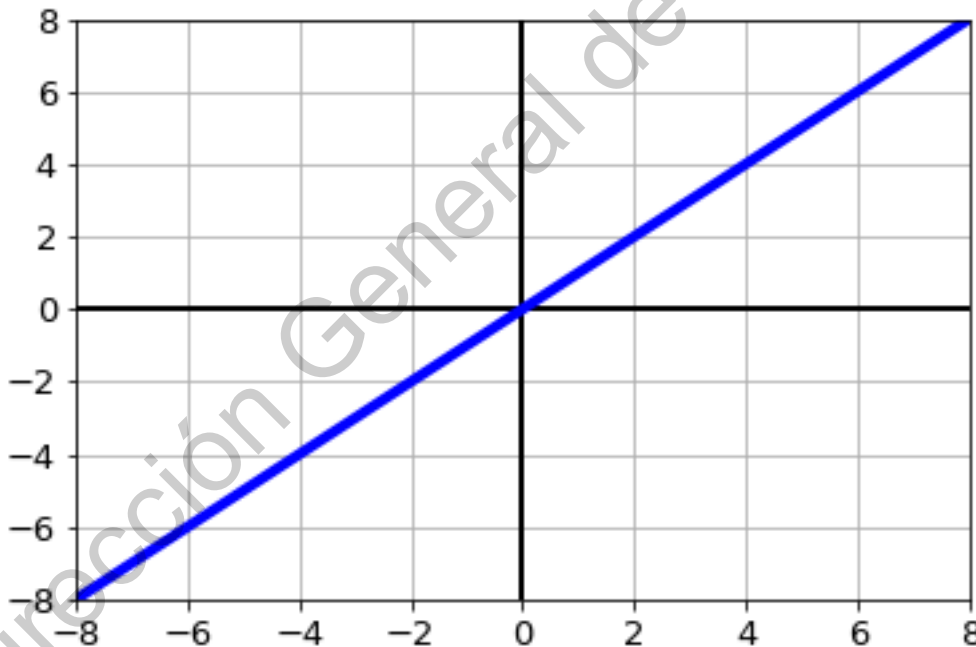


Figura 3.10. Función de activación lineal. Es una de las funciones de activación más simples, es rápido de calcular y no tiene restricciones en la salida.

Uno de sus beneficios es que los valores negativos pueden ser manejados en la CNN, algo que puede ser útil para el método de normalización seleccionado que se aplica aquí.

3.5.1.2. Unidad lineal rectificada (ReLU)

Esta función de activación es una variación de la función de activación lineal, la principal diferencia es que los valores negativos se transforman en ceros, y los valores positivos se pasan sin transformación. En la ecuación 3.3 y la Figura 3.11 se muestra su comportamiento.

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (3.3)$$

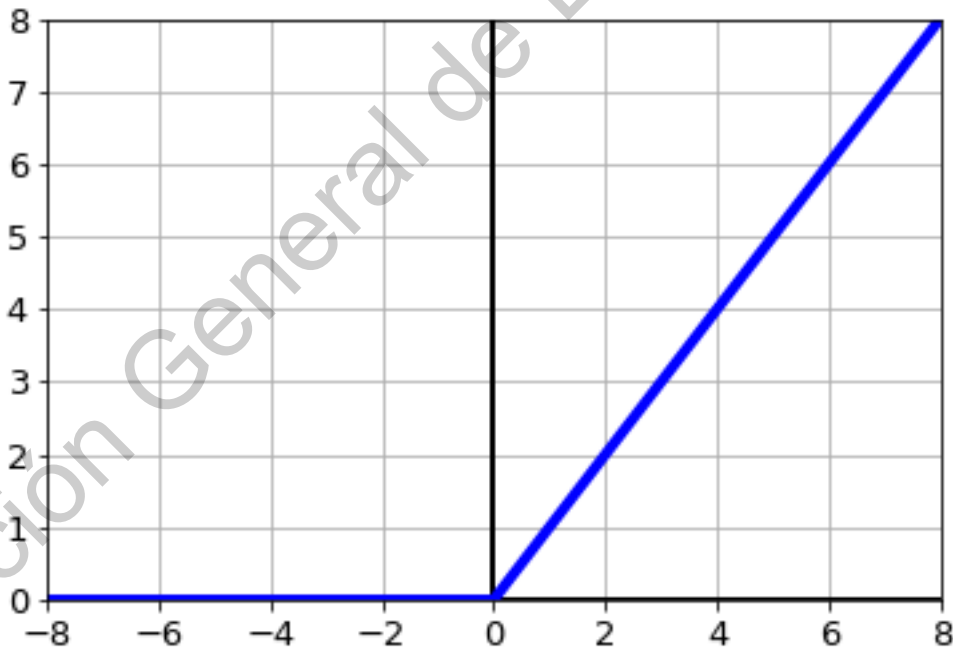


Figura 3.11. Función de activación ReLU. Esta función de activación también es rápida de calcular, pero permite una transformación de los datos de entrada.

El entrenamiento de redes neuronales profundas con ReLU y un ReLU parametrizado (PReLU) ha sido estudiado en (Arora et al., 2018; He et al., 2015), ambos haciendo investigación en una perspectiva teórica del desempeño de esas funciones de activación.

3.5.1.3. Sigmoide

La función de activación sigmoide tiene una salida que oscila entre 0 y 1. En la ecuación 3.4 y la Figura 3.12 se muestra su comportamiento.

$$f(x) = \frac{1}{1+e^{-x}} \quad (3.4)$$

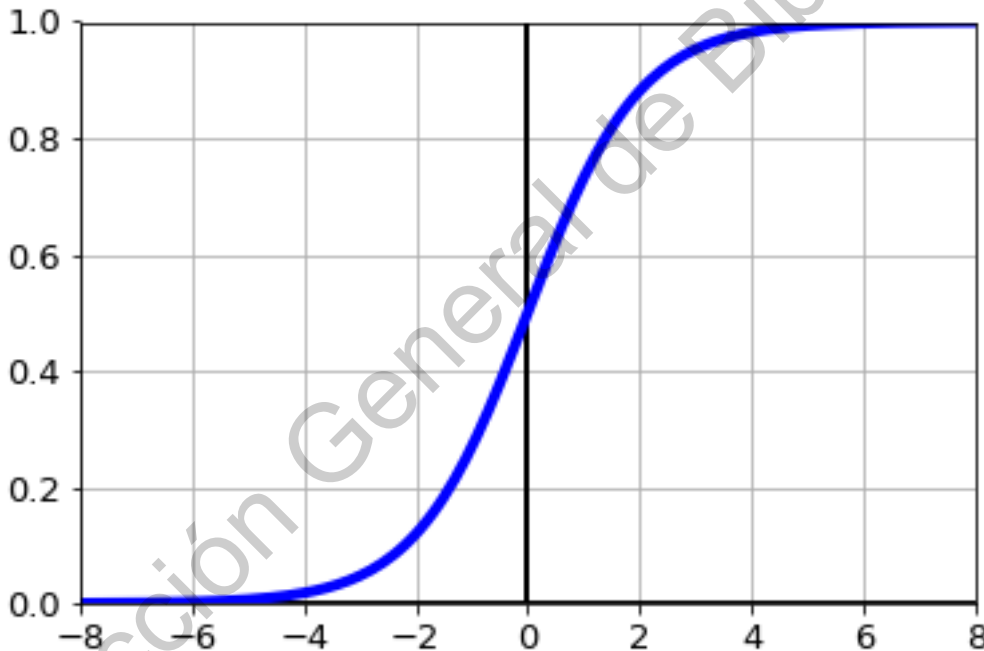


Figura 3.12. Función de activación sigmoide. Esta función de activación es muy adecuada para problemas de clasificación, específicamente para la clasificación binaria. Es más lento de calcular en comparación con las funciones ReLU y de activación lineal.

3.5.1.4. Softplus

La función de activación softplus tiene una salida superior a 0, es similar a ReLU con la diferencia de que softplus tiene una adaptación menos abrupta sobre los valores en los límites cercanos a 0. En la ecuación 3.5 y en la Figura 3.13 se muestra su comportamiento.

$$f(x) = \ln(1 + e^x) \quad (3.5)$$

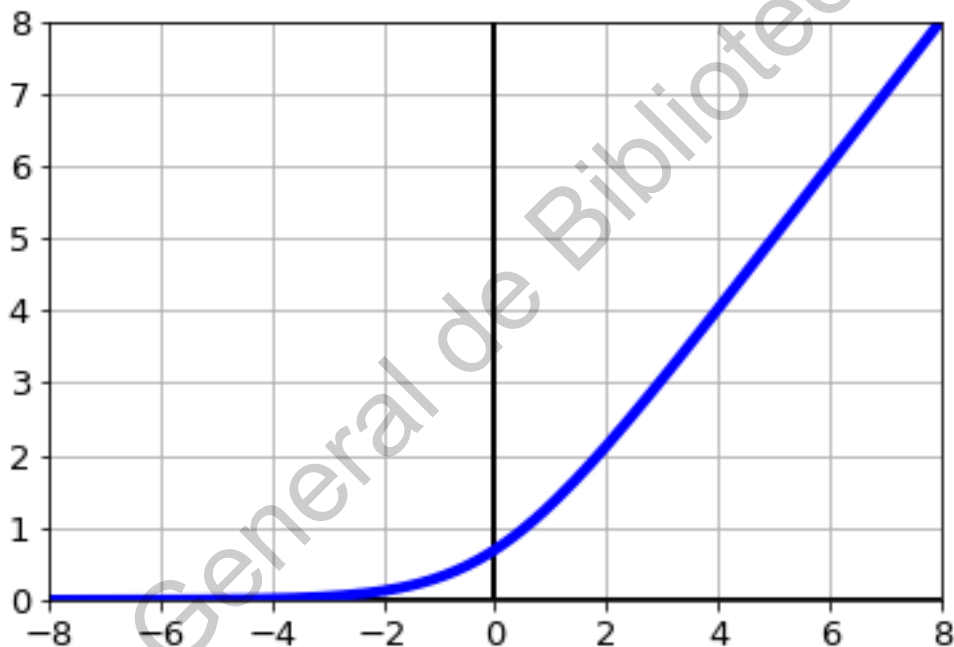


Figura 3.13. Función de activación Softplus. Este también es más lento de calcular que el ReLU y la función de activación lineal, pero es muy adecuado para aplicaciones de regresión y no tiene un cambio drástico en los valores de salida como el ReLU.

3.5.2. Optimizador

Los optimizadores son algoritmos que permiten que las redes neuronales tengan un mejor rendimiento en cada iteración. Se basan en determinados criterios en función del optimizador elegido. Los optimizadores determinan cómo se

actualizarán los pesos de cada capa, por lo que son una parte crucial en el rendimiento de una red neuronal.

3.5.2.1. Adam

Adam es uno de los optimizadores más populares hoy en día. Este optimizador se basa en una estimación del momento adaptativo. Adam es "muy adecuado para problemas grandes en términos de datos y/o parámetros. El método también es apropiado para objetivos no estacionarios y problemas con gradientes muy ruidosos y/o dispersos" (Kingma y Ba, 2015, p.1)

3.5.3. Tamaño del kernel

Este hiperparámetro es exclusivo de CNN porque define el núcleo de convolución que es la característica principal de esta arquitectura. Consiste en una matriz bidimensional con valores diferentes en cada celda. Cuando se aplica una capa de convolución, el primer paso es la creación de *kernels*. No hay ninguna regla para determinar el número de *kernels* a utilizar ya que esto queda a consideración del diseñador del modelo. Revisando el desarrollo de arquitecturas populares de CNN, podemos determinar que un mayor número de *kernels*, en general, es mejor.

Solo hace falta revisar las arquitecturas ganadoras del "*ImageNet Large Scale Visual Recognition Competition*" (ILSVRC), que es una de las competencias más populares de visión computacional en la que las CNN han demostrado, en los últimos años, un gran desempeño. En 2012 la red AlexNet, con 1,376 *kernels* entre todas sus capas, ganó esta competencia (Krizhevsky et al., 2012). En 2014, Google LeNET ganó con una suma que resulta en 5744 *kernels* (Szegedy et al., 2015). En 2016 la arquitectura ganadora fue ResNeXt (Xie et al., 2017), con 8,192 *kernels*. Por último, una de las arquitecturas más populares para detección de objetos llamada

YOLO (You Only Look Once), está inspirada en Google LeNET y cuenta con aproximadamente 13,000 *kernels* en total (Redmon et al., 2016).

Con el proceso de aplicación de *kernels*, se realiza un gran número de transformaciones a los datos. Aquí es donde se extraen las características mediante la combinación de las variables de entrada. El tamaño del núcleo tiene un efecto en el proceso. Con núcleos más pequeños la combinación es con las células más cercanas. Por otro lado, los núcleos más grandes pueden resultar en una mala representación de los datos ya que generalizan un número muy elevado de celdas.

3.6. Métodos de Ensamble

La idea principal de los métodos de ensamble es determinar una predicción más precisa mediante el voto de diversos modelos. Esto ha sido estudiado en (Syarif et al., 2012; Graczyk et al. 2010; Berk et al., 2006) donde se concluyó que se obtiene un modelo más general para la predicción cuando se aplica un método de conjunto, pero no en todas las aplicaciones.

El *bagging* es uno de los métodos de ensamble más populares. Consiste en la creación de diversos modelos entrenados, luego asignando una importancia de voto igual a cada modelo para finalmente calcular la predicción del conjunto (Figura 3.14). Esto se obtiene mediante el cálculo de la media de las predicciones de cada modelo individual del conjunto.

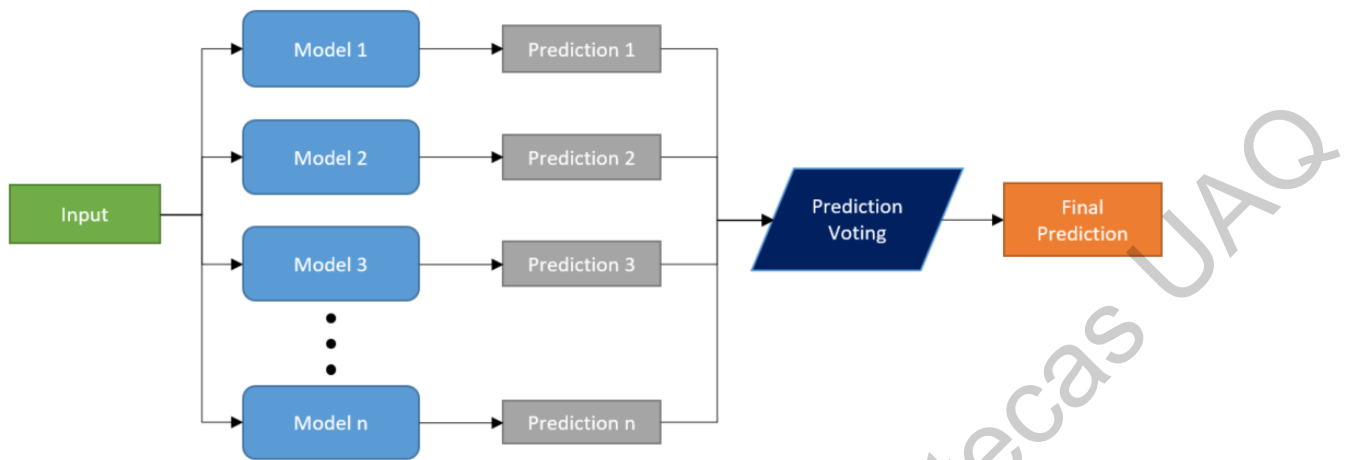


Figura 3.14. Diagrama del conjunto de Bagging.

Hay 4 etapas en el proceso de *bagging* como se muestra en la figura 3.14. Al principio, es la entrada a cada modelo en el conjunto. A continuación, se calcula el resultado de cada modelo; seguido del proceso de votación para obtener finalmente la predicción del conjunto.

En comparación con otros métodos de ensamble, *bagging* es más consistente con los resultados (Graczyk et al. 2010). El método de *stacking*, por ejemplo, produce resultados variables altos, en algunos casos aumentando la precisión de la predicción, y en otros casos dando una predicción inexacta incluso si todos los modelos del conjunto tienen una buena precisión de predicción. Por el contrario, el *staking* proporciona un mejor rendimiento de precisión la mayoría de las veces, incluso si se trata de una ligera mejora (Graczyk et al. 2010).

3.7. Métricas de cálculo del error del modelo

El uso de métricas confiables es esencial para sondear el desempeño de los resultados. También es importante tener métricas comúnmente utilizadas con el propósito de hacer comparaciones con otros trabajos. En este trabajo, las métricas

utilizadas son: error cuadrático medio de la raíz, error porcentual absoluto medio, índice de acuerdo y coeficiente de determinación.

3.7.1. Error cuadrático medio (RMSE)

Esta métrica es una de las más utilizadas en el análisis de regresión. Es indiferente a la dirección de los errores, esto significa que va de 0 a infinito. La fórmula para obtener la RMSE se muestra en la ecuación 3.6.

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2} \quad (3.6)$$

Dónde:

n - Número de observaciones.

y_j - Observación.

y[^]_j - Predicción.

3.7.2. Error de porcentaje absoluto medio (MAPE)

MAPE presenta el desempeño de la predicción en relación con el valor esperado. Esta métrica da un valor porcentual que facilita su interpretación, pero tiene el inconveniente de ser volátil en función de la magnitud de los valores utilizados. El rango de esta métrica es de 0 a 1, siendo 1 una puntuación perfecta. La fórmula para obtener el MAPE se muestra en la ecuación 3.7.

$$MAPE = \frac{1}{n} \sum_{j=1}^n \left| \frac{y_j - \hat{y}_j}{y_j} \right| \quad (3.7)$$

Dónde:

n - Número de observaciones.

y_j - Observación.

$y^{\wedge}j$ - Predicción.

3.7.3. Índice de acuerdo (IOA)

Esta métrica también utiliza el error cuadrado, pero añade un factor en el que tanto el valor esperado como la predicción se comparan con una medición de tendencia central del conjunto de datos (media). Estas métricas van de 0 a 1, siendo 1 la combinación perfecta entre la observación y la predicción. La fórmula para obtener el IOA se muestra en la ecuación 3.8.

$$IOA = 1 - \frac{\sum_{j=1}^n (y_j - \hat{y}_j)^2}{\sum_{j=1}^n (|\hat{y}_j - \mu_T| + |y_j - \mu_T|)^2} \quad (3.8)$$

Dónde:

n - Número de observaciones.

y_j - Observación.

$y^{\wedge}j$ - Predicción.

μ_T - Media del conjunto de datos.

3.7.4. Coeficiente de determinación (R2)

El uso de R2 está bien establecido para el análisis de regresión. Define la proporción de la varianza en el modelo, haciéndola útil para medir las capacidades predictivas de un modelo (Nagelkerke, 1991). La diferencia con el coeficiente de correlación (R) es que R2 presenta un rango de valores de 0 a 1 en lugar de -1 a 1. Esto se debe a que R2 da la misma puntuación a la correlación negativa y a la correlación positiva entre variables de la misma magnitud.

La fórmula para obtener el coeficiente de determinación se muestra en la ecuación 3.9.

$$R^2 = \left(\frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \right)^2 \quad (3.9)$$

Dónde:

n - Número de observaciones.

x_i - Observación de la variable 1.

\bar{x} - Predicción de la variable 1.

y_i - Observación de la variable 2.

\bar{y} - Predicción de la variable 2.

Dirección General de Bibliotecas UAQ

4. Resultados

En este estudio se probaron varias arquitecturas para MLP y CNN. Para cada arquitectura se interpretó un conjunto con todas las variantes creadas. Además, se determinó el mejor conjunto de todo el conjunto de arquitecturas. Para obtener el mejor conjunto se ha tenido en cuenta el RMSE (Root Mean Square Error), por lo que el algoritmo consistía en descartar una arquitectura y calcular el RMSE del conjunto resultante hasta que las arquitecturas elegidas mostraran un RMSE superior. En la Tabla 4.1 se presentan los resultados para cada combinación de MLP y el conjunto de todas las arquitecturas.

Tabla 4.1. Resultados de arquitecturas MLP y conjunto completo.

Modelo	RMSE	MAPE	IOA	R2
MLP_AF_linear_DN_2	19.7712	0.2385	0.8686	0.5863
MLP_AF_linear_DN_4	19.764	0.2378	0.8707	0.5866
MLP_AF_linear_DN_6	19.7841	0.2365	0.868	0.5858
MLP_AF_linear_DN_8	19.7649	0.2366	0.8705	0.5866
MLP_AF_linear_DN_10	19.758	0.2369	0.8713	0.5869
MLP_AF_linear_DN_12	19.7803	0.2379	0.8732	0.586
MLP_AF_linear_DN_14	19.7654	0.2361	0.8632	0.5866
MLP_AF_linear_DN_16	19.7369	0.2353	0.8673	0.5878
MLP_AF_linear_DN_18	19.7995	0.2384	0.8731	0.5851
MLP_AF_linear_DN_20	19.716	0.2352	0.8698	0.5886
MLP_AF_relu_DN_2	19.8538	0.2385	0.8689	0.5829
MLP_AF_relu_DN_4	20.4308	0.2409	0.8582	0.5583
MLP_AF_relu_DN_6	20.4799	0.246	0.8528	0.5561
MLP_AF_relu_DN_8	20.5451	0.2501	0.8559	0.5533
MLP_AF_relu_DN_10	20.4915	0.2516	0.8581	0.5556
MLP_AF_relu_DN_12	20.3268	0.2403	0.8579	0.5628
MLP_AF_relu_DN_14	20.3872	0.2459	0.858	0.5601
MLP_AF_relu_DN_16	20.5656	0.2522	0.8518	0.5524

Tabla 4.1 (cont). Resultados de arquitecturas MLP y conjunto completo.

Modelo	RMSE	MAPE	IOA	R2
MLP_AF_relu_DN_18	20.3962	0.2476	0.8575	0.5598
MLP_AF_relu_DN_20	20.4083	0.248	0.8574	0.5592
MLP_AF_sigmoid_DN_2	20.5594	0.2506	0.8595	0.5527
MLP_AF_sigmoid_DN_4	20.5473	0.2546	0.861	0.5532
MLP_AF_sigmoid_DN_6	20.4292	0.2518	0.8644	0.5583
MLP_AF_sigmoid_DN_8	20.3239	0.248	0.8628	0.5629
MLP_AF_sigmoid_DN_10	20.4508	0.2523	0.862	0.5574
MLP_AF_sigmoid_DN_12	20.2373	0.2501	0.8657	0.5666
MLP_AF_sigmoid_DN_14	20.162	0.2435	0.8654	0.5698
MLP_AF_sigmoid_DN_16	20.184	0.2455	0.8666	0.5689
MLP_AF_sigmoid_DN_18	20.2226	0.2496	0.8646	0.5672
MLP_AF_sigmoid_DN_20	20.1727	0.2476	0.8664	0.5694
MLP_AF_softplus_DN_2	20.4178	0.2449	0.8457	0.5588
MLP_AF_softplus_DN_4	20.14	0.2415	0.8599	0.5708
MLP_AF_softplus_DN_6	20.0486	0.2378	0.862	0.5746
MLP_AF_softplus_DN_8	19.9937	0.2387	0.8628	0.577
MLP_AF_softplus_DN_10	20.0855	0.2403	0.8626	0.5731
MLP_AF_softplus_DN_12	20.061	0.2397	0.8657	0.5741
MLP_AF_softplus_DN_14	20.0226	0.2375	0.8639	0.5757
MLP_AF_softplus_DN_16	20.1624	0.247	0.8612	0.5698
MLP_AF_softplus_DN_18	20.0142	0.2405	0.8641	0.5761
MLP_AF_softplus_DN_20	20.1047	0.2408	0.8602	0.5723
Ensamble	19.9617	0.2405	0.8652	0.5783

Se presentan las métricas para cada una de las arquitecturas MLP. El nombre en cada fila representa los parámetros seleccionados para esa arquitectura en particular separados por guiones bajos. El nombre comienza con MLP seguido de AF que significa "función de activación". Después de esto está la función de activación seleccionada. A continuación, se puede encontrar DN que significa "neuronas de capa densa". Finalmente está el número de neuronas seleccionadas.

En la parte inferior de la tabla están los resultados del ensamble de todas las arquitecturas. En las tablas de cada tipo de red utilizada se señalan las tres mejores arquitecturas con negrita.

Una vez realizada la evaluación de cada arquitectura, se obtuvo un ensamble con arquitecturas seleccionadas para el conjunto. Los resultados se presentan en la Tabla 4.2. En dicha tabla se presentan los modelos que forman parte del ensamble y los resultados obtenidos por dicho ensamble

Tabla 4.2. Modelos que forman parte del ensamble de arquitecturas MLP y resultado del ensamble.

Ensamble	RMSE	MAPE	IOA	R2
MLP_AF_linear_DN_10				
MLP_AF_linear_DN_12				
MLP_AF_linear_DN_14	19.7213	0.239	0.863	0.582
MLP_AF_linear_DN_16				
MLP_AF_linear_DN_20				
MLP_AF_softplus_DN_14				

Las arquitecturas que presentaron un mejor desempeño combinado se presentan aquí, junto a las métricas del conjunto. El nombre se define en el mismo sentido de la descripción de la Tabla 4.1. La función de activación lineal es la más repetida en este conjunto.

En la Tabla 4.3 se presentan los resultados para cada combinación de 1DCNN con una capa convolucional y el conjunto de todas las arquitecturas.

Tabla 4.3. Resultados de 1DCNN con arquitecturas de una capa convolucional y conjunto completo.

Modelo	RMSE	MAPE	IOA	R²
CNN_AF_linear_KS_2	19.3736	0.2395	0.8782	0.6028
CNN_AF_linear_KS_3	19.3865	0.2365	0.8725	0.6023
CNN_AF_linear_KS_4	19.4665	0.2371	0.8664	0.599
CNN_AF_linear_KS_5	19.4786	0.2353	0.8641	0.5985
CNN_AF_linear_KS_6	19.4022	0.2371	0.8695	0.6016
CNN_AF_linear_KS_7	19.5374	0.24	0.8683	0.5961
CNN_AF_relu_KS_2	22.1398	0.2584	0.8247	0.4813
CNN_AF_relu_KS_3	21.6551	0.2639	0.842	0.5037
CNN_AF_relu_KS_4	22.7304	0.2884	0.8316	0.4532
CNN_AF_relu_KS_5	22.4147	0.2734	0.8334	0.4683
CNN_AF_relu_KS_6	22.5424	0.2586	0.8309	0.4622
CNN_AF_relu_KS_7	22.7235	0.2786	0.8158	0.4536
CNN_AF_sigmoid_KS_2	20.1312	0.2581	0.8713	0.5711
CNN_AF_sigmoid_KS_3	20.0372	0.2333	0.8749	0.5751
CNN_AF_sigmoid_KS_4	20.8611	0.2741	0.8601	0.5395
CNN_AF_sigmoid_KS_5	21.3648	0.2602	0.8513	0.517
CNN_AF_sigmoid_KS_6	21.5005	0.2659	0.8493	0.5108
CNN_AF_sigmoid_KS_7	22.5766	0.2845	0.8386	0.4606
CNN_AF_softplus_KS_2	19.6784	0.2289	0.8743	0.5902
CNN_AF_softplus_KS_3	20.2479	0.24	0.8719	0.5661
CNN_AF_softplus_KS_4	21.1348	0.2619	0.8594	0.5273
CNN_AF_softplus_KS_5	21.3735	0.2718	0.8474	0.5166
CNN_AF_softplus_KS_6	21.4527	0.2752	0.8546	0.513
CNN_AF_softplus_KS_7	22.1175	0.2666	0.8344	0.4823
Ensamble	19.5057	0.2372	0.8704	0.5974

El nombre en cada fila es similar al de las Tablas 4.1 y 4.2. La diferencia es que en esta tabla los nombres comienzan con CNN porque es la arquitectura utilizada. El nombre continúa con AF (función de activación), seguido de la función de

activación seleccionada para esa arquitectura. Finalmente, el nombre termina con KS (tamaño del núcleo) seguido del tamaño del núcleo unidimensional utilizado. Las tres mejores arquitecturas están en negrita.

Las arquitecturas resultantes que presentaron el mejor desempeño combinado se presentan como un ensamble en la Tabla 4.4, junto a la puntuación de cada métrica.

Tabla 4.4. Ensamble final para 1DCNN con arquitecturas de una capa convolucional.

Modelo	RMSE	MAPE	IOA	R²
CNN_AF_linear_KS_2				
CNN_AF_linear_KS_5				
CNN_AF_linear_KS_6				
CNN_AF_sigmoid_KS_3	19.0053	0.2315	0.8805	0.624
CNN_AF_softplus_KS_2				
CNN_AF_softplus_KS_3				

En esta tabla se presenta el ensamble final de 1DCNN con una capa convolucional que contiene las arquitecturas que presentaron el mejor rendimiento combinado. Se observa que la mayoría de las arquitecturas tienen la función de activación lineal.

En la Tabla 4.5 se presentan los resultados para cada combinación de 1DCNN con dos capas convolucionales y el conjunto de todas las arquitecturas.

Tabla 4.5. Resultados de 1DCNN con dos arquitecturas de capas convolucionales y conjunto completo.

Modelo	RMSE	MAPE	IOA	R2
CNN_AF_linear_KS1_2_KS2_2	19.3009	0.233	0.8723	0.6058
CNN_AF_linear_KS1_2_KS2_3	19.4228	0.237	0.8752	0.6008
CNN_AF_linear_KS1_2_KS2_4	19.5803	0.2417	0.8668	0.5943
CNN_AF_linear_KS1_2_KS2_5	19.5858	0.2376	0.8604	0.594
CNN_AF_linear_KS1_2_KS2_6	19.3982	0.2319	0.8635	0.6018
CNN_AF_linear_KS1_3_KS2_2	19.6809	0.2499	0.8696	0.5901
CNN_AF_linear_KS1_3_KS2_3	19.46	0.2404	0.8726	0.5992
CNN_AF_linear_KS1_3_KS2_4	19.6259	0.2404	0.8629	0.5924
CNN_AF_linear_KS1_3_KS2_5	19.5139	0.2432	0.8759	0.597
CNN_AF_linear_KS1_4_KS2_2	19.6161	0.2356	0.8554	0.5928
CNN_AF_linear_KS1_4_KS2_3	19.3451	0.2348	0.8745	0.604
CNN_AF_linear_KS1_4_KS2_4	19.8916	0.2603	0.8805	0.5813
CNN_AF_linear_KS1_5_KS2_2	19.6789	0.2363	0.8575	0.5902
CNN_AF_linear_KS1_5_KS2_3	19.5545	0.2421	0.8741	0.5953
CNN_AF_linear_KS1_6_KS2_2	19.701	0.2533	0.8803	0.5893
CNN_AF_relu_KS1_2_KS2_2	22.9559	0.2676	0.8408	0.4423
CNN_AF_relu_KS1_2_KS2_3	23.6301	0.2924	0.8255	0.4091
CNN_AF_relu_KS1_2_KS2_4	23.4664	0.2814	0.8066	0.4172
CNN_AF_relu_KS1_2_KS2_5	23.1857	0.2649	0.8229	0.4311
CNN_AF_relu_KS1_2_KS2_6	22.9636	0.2731	0.8346	0.442
CNN_AF_relu_KS1_3_KS2_2	22.9772	0.2676	0.8337	0.4413
CNN_AF_relu_KS1_3_KS2_3	22.2487	0.2544	0.8502	0.4762
CNN_AF_relu_KS1_3_KS2_4	22.2388	0.2716	0.8364	0.4766
CNN_AF_relu_KS1_3_KS2_5	22.8021	0.2667	0.8299	0.4498
CNN_AF_relu_KS1_4_KS2_2	23.2384	0.2885	0.8289	0.4285
CNN_AF_relu_KS1_4_KS2_3	23.0337	0.2687	0.8229	0.4385
CNN_AF_relu_KS1_4_KS2_4	23.7178	0.2945	0.8196	0.4047
CNN_AF_relu_KS1_5_KS2_2	23.5464	0.283	0.8157	0.4133
CNN_AF_relu_KS1_5_KS2_3	23.0881	0.2708	0.8164	0.4359
CNN_AF_relu_KS1_6_KS2_2	22.7216	0.2937	0.8333	0.4537
CNN_AF_sigmoid_KS1_2_KS2_2	20.8244	0.2399	0.8792	0.5411
CNN_AF_sigmoid_KS1_2_KS2_3	20.2936	0.239	0.8781	0.5642
CNN_AF_sigmoid_KS1_2_KS2_4	20.3843	0.2407	0.8782	0.5603
CNN_AF_sigmoid_KS1_2_KS2_5	20.8016	0.2536	0.8687	0.5421
CNN_AF_sigmoid_KS1_2_KS2_6	20.258	0.2524	0.8714	0.5657
CNN_AF_sigmoid_KS1_3_KS2_2	21.4882	0.2439	0.8665	0.5114

Tabla 4.5 (cont). Resultados de 1DCNN con dos arquitecturas de capas convolucionales y conjunto completo.

Modelo	RMSE	MAPE	IOA	R2
CNN_AF_sigmoid_KS1_3_KS2_3	21.0043	0.2698	0.8703	0.5331
CNN_AF_sigmoid_KS1_3_KS2_4	20.9379	0.2509	0.8699	0.5361
CNN_AF_sigmoid_KS1_3_KS2_5	21.6733	0.2755	0.8533	0.5029
CNN_AF_sigmoid_KS1_4_KS2_2	21.8593	0.2727	0.8527	0.4943
CNN_AF_sigmoid_KS1_4_KS2_3	21.6396	0.2657	0.8514	0.5045
CNN_AF_sigmoid_KS1_4_KS2_4	21.3357	0.2576	0.856	0.5183
CNN_AF_sigmoid_KS1_5_KS2_2	21.9651	0.2788	0.8548	0.4894
CNN_AF_sigmoid_KS1_5_KS2_3	21.5466	0.2808	0.857	0.5087
CNN_AF_sigmoid_KS1_6_KS2_2	22.3244	0.2821	0.8434	0.4726
CNN_AF_softplus_KS1_2_KS2_2	20.18	0.2463	0.8603	0.569
CNN_AF_softplus_KS1_2_KS2_3	21.4367	0.2583	0.8592	0.5137
CNN_AF_softplus_KS1_2_KS2_4	21.1607	0.2661	0.8542	0.5261
CNN_AF_softplus_KS1_2_KS2_5	20.8701	0.2499	0.8543	0.5391
CNN_AF_softplus_KS1_2_KS2_6	20.7139	0.2688	0.8565	0.5459
CNN_AF_softplus_KS1_3_KS2_2	20.7989	0.2503	0.8639	0.5422
CNN_AF_softplus_KS1_3_KS2_3	20.8405	0.2513	0.8561	0.5404
CNN_AF_softplus_KS1_3_KS2_4	21.5592	0.2619	0.8579	0.5081
CNN_AF_softplus_KS1_3_KS2_5	21.445	0.2551	0.8645	0.5133
CNN_AF_softplus_KS1_4_KS2_2	21.5239	0.2587	0.8447	0.5097
CNN_AF_softplus_KS1_4_KS2_3	21.6114	0.2657	0.8395	0.5057
CNN_AF_softplus_KS1_4_KS2_4	21.0746	0.2513	0.8559	0.53
CNN_AF_softplus_KS1_5_KS2_2	22.0445	0.2528	0.8461	0.4857
CNN_AF_softplus_KS1_5_KS2_3	21.885	0.2828	0.8448	0.4931
CNN_AF_softplus_KS1_6_KS2_2	21.1222	0.2601	0.8545	0.5279
Ensamble	19.3111	0.2336	0.8757	0.6054

En esta tabla, los nombres de cada fila se estructuran de la siguiente manera. Comienza con CNN para la arquitectura convolucional implementada. A continuación, las letras AF (función de activación) seguidas de la función de activación. Finalmente, KS1 (tamaño del núcleo de la primera capa convolucional) y KS2 (tamaño del núcleo de la segunda capa convolucional) se muestran seguidos por el tamaño del núcleo seleccionado respectivamente. Cada parámetro está separado por un guion bajo. Las tres mejores arquitecturas están en negrita.

El ensamble resultante de 1DCNN con dos capas convolucionales es bastante diverso. Contiene al menos dos arquitecturas de cada función de activación, pero hay que tener en cuenta que la mayoría de los tamaños de núcleo en ambas capas son pequeños. En la Tabla 4.6 se presenta este conjunto.

Tabla 4.6. Ensamble final para 1DCNN con dos arquitecturas de capas convolucionales.

Modelo	RMSE	MAPE	IOA	R2
CNN_AF_linear_KS1_2_KS2_2				
CNN_AF_linear_KS1_2_KS2_3				
CNN_AF_linear_KS1_2_KS2_5				
CNN_AF_linear_KS1_2_KS2_6				
CNN_AF_linear_KS1_4_KS2_2				
CNN_AF_linear_KS1_5_KS2_2				
CNN_AF_relu_KS1_3_KS2_3	18.9024	0.215	0.8762	0.6093
CNN_AF_relu_KS1_6_KS2_2				
CNN_AF_sigmoid_KS1_2_KS2_2				
CNN_AF_sigmoid_KS1_2_KS2_3				
CNN_AF_sigmoid_KS1_2_KS2_4				
CNN_AF_softplus_KS1_2_KS2_2				
CNN_AF_softplus_KS1_2_KS2_3				

Este ensamble presentó una mejora en comparación con el ensamble de MLP y el ensamble de 1DCNN con una capa de convolución. Contiene una mayor diversidad en las funciones de activación, pero tiende a tener pequeños tamaños de núcleo.

En la Tabla 4.7 se presentan los resultados de cada combinación de 2DCNN y el ensamble de todas las arquitecturas.

Tabla 4.7. Resultados de las arquitecturas 2DCNN y ensamble completo.

Modelo	RMSE	MAPE	IOA	R2
CNN2D_AF_linear_KSI_10_KSF_2	19.4114	0.2345	0.8671	0.6012
CNN2D_AF_linear_KSI_20_KSF_2	19.5741	0.24	0.8692	0.5945
CNN2D_AF_linear_KSI_30_KSF_2	19.6173	0.2467	0.8794	0.5927
CNN2D_AF_linear_KSI_40_KSF_2	19.8276	0.2484	0.8803	0.584
CNN2D_AF_linear_KSI_50_KSF_2	19.4803	0.2372	0.8678	0.5984
CNN2D_AF_linear_KSI_10_KSF_3	19.4846	0.2381	0.8727	0.5982
CNN2D_AF_linear_KSI_20_KSF_3	19.2497	0.2314	0.8704	0.6079
CNN2D_AF_linear_KSI_30_KSF_3	19.8524	0.2386	0.8501	0.5829
CNN2D_AF_linear_KSI_40_KSF_3	20.4432	0.2549	0.8835	0.5577
CNN2D_AF_linear_KSI_50_KSF_3	19.3115	0.2338	0.8729	0.6053
CNN2D_AF_linear_KSI_10_KSF_4	19.6005	0.2418	0.8716	0.5934
CNN2D_AF_linear_KSI_20_KSF_4	19.6324	0.239	0.8625	0.5921
CNN2D_AF_linear_KSI_30_KSF_4	19.7419	0.2453	0.8816	0.5876
CNN2D_AF_linear_KSI_40_KSF_4	19.6444	0.2415	0.8656	0.5916
CNN2D_AF_linear_KSI_50_KSF_4	19.4299	0.2345	0.8653	0.6005
CNN2D_AF_linear_KSI_10_KSF_5	19.5017	0.2379	0.865	0.5975
CNN2D_AF_linear_KSI_20_KSF_5	19.8119	0.2561	0.874	0.5846
CNN2D_AF_linear_KSI_30_KSF_5	19.5369	0.2418	0.8777	0.5961
CNN2D_AF_linear_KSI_40_KSF_5	19.4853	0.2428	0.8771	0.5982
CNN2D_AF_linear_KSI_50_KSF_5	19.527	0.2318	0.8567	0.5965
CNN2D_AF_linear_KSI_10_KSF_6	19.6254	0.2411	0.8801	0.5924
CNN2D_AF_linear_KSI_20_KSF_6	19.4672	0.2354	0.8619	0.599
CNN2D_AF_linear_KSI_30_KSF_6	19.4265	0.2387	0.8676	0.6006
CNN2D_AF_linear_KSI_40_KSF_6	19.5761	0.2385	0.8647	0.5945
CNN2D_AF_linear_KSI_50_KSF_6	19.2326	0.2317	0.8747	0.6086
CNN2D_AF_linear_KSI_10_KSF_7	19.6593	0.2418	0.8622	0.591
CNN2D_AF_linear_KSI_20_KSF_7	19.5188	0.2396	0.868	0.5968
CNN2D_AF_linear_KSI_30_KSF_7	20.1355	0.2569	0.8802	0.5709
CNN2D_AF_linear_KSI_40_KSF_7	19.8201	0.2426	0.8584	0.5843
CNN2D_AF_linear_KSI_50_KSF_7	19.7392	0.2395	0.8652	0.5877
CNN2D_AF_relu_KSI_10_KSF_2	21.8331	0.263	0.8369	0.4955
CNN2D_AF_relu_KSI_20_KSF_2	22.09	0.263	0.8239	0.4836
CNN2D_AF_relu_KSI_30_KSF_2	22.744	0.2538	0.8329	0.4526
CNN2D_AF_relu_KSI_40_KSF_2	21.9948	0.2567	0.82	0.488
CNN2D_AF_relu_KSI_50_KSF_2	22.189	0.2585	0.8476	0.479
CNN2D_AF_relu_KSI_10_KSF_3	22.6045	0.274	0.8019	0.4593
CNN2D_AF_relu_KSI_20_KSF_3	21.6522	0.2512	0.8303	0.5039

Tabla 4.7 (cont). Resultados de las arquitecturas 2DCNN y ensamble completo.

Modelo	RMSE	MAPE	IOA	R2
CNN2D_AF_relu_KSI_30_KSF_3	22.063	0.2598	0.8289	0.4849
CNN2D_AF_relu_KSI_40_KSF_3	22.1329	0.239	0.8258	0.4816
CNN2D_AF_relu_KSI_50_KSF_3	22.0719	0.2602	0.8375	0.4845
CNN2D_AF_relu_KSI_10_KSF_4	21.59	0.2437	0.8275	0.5067
CNN2D_AF_relu_KSI_20_KSF_4	21.6647	0.2515	0.8178	0.5033
CNN2D_AF_relu_KSI_30_KSF_4	21.8427	0.2576	0.8442	0.4951
CNN2D_AF_relu_KSI_40_KSF_4	21.8096	0.2468	0.8232	0.4966
CNN2D_AF_relu_KSI_50_KSF_4	22.0503	0.2552	0.8436	0.4855
CNN2D_AF_relu_KSI_10_KSF_5	22.3912	0.2645	0.8213	0.4694
CNN2D_AF_relu_KSI_20_KSF_5	21.4918	0.2465	0.827	0.5112
CNN2D_AF_relu_KSI_30_KSF_5	21.6832	0.2471	0.8392	0.5025
CNN2D_AF_relu_KSI_40_KSF_5	21.9802	0.2628	0.8478	0.4887
CNN2D_AF_relu_KSI_50_KSF_5	22.1293	0.2646	0.8336	0.4818
CNN2D_AF_relu_KSI_10_KSF_6	21.8834	0.2526	0.8417	0.4932
CNN2D_AF_relu_KSI_20_KSF_6	22.6163	0.2793	0.81	0.4587
CNN2D_AF_relu_KSI_30_KSF_6	22.1687	0.2595	0.8317	0.4799
CNN2D_AF_relu_KSI_40_KSF_6	22.3048	0.2936	0.8272	0.4735
CNN2D_AF_relu_KSI_50_KSF_6	23.1206	0.281	0.8121	0.4343
CNN2D_AF_relu_KSI_10_KSF_7	22.3499	0.2554	0.8472	0.4714
CNN2D_AF_relu_KSI_20_KSF_7	22.5012	0.262	0.8291	0.4642
CNN2D_AF_relu_KSI_30_KSF_7	22.4501	0.2611	0.8347	0.4666
CNN2D_AF_relu_KSI_40_KSF_7	21.8609	0.2596	0.8435	0.4943
CNN2D_AF_relu_KSI_50_KSF_7	22.7218	0.2706	0.8232	0.4536
CNN2D_AF_sigmoid_KSI_10_KSF_2	34.0586	0.4265	0.3728	0.2276
CNN2D_AF_sigmoid_KSI_20_KSF_2	34.0407	0.426	0.3724	0.2263
CNN2D_AF_sigmoid_KSI_30_KSF_2	33.8802	0.4212	0.3685	0.2147
CNN2D_AF_sigmoid_KSI_40_KSF_2	25.4441	0.3619	0.7632	0.3149
CNN2D_AF_sigmoid_KSI_50_KSF_2	20.1622	0.2429	0.8708	0.5698
CNN2D_AF_sigmoid_KSI_10_KSF_3	33.9219	0.4224	0.3695	0.2177
CNN2D_AF_sigmoid_KSI_20_KSF_3	33.7903	0.4185	0.3662	0.2083
CNN2D_AF_sigmoid_KSI_30_KSF_3	33.8601	0.4206	0.368	0.2133
CNN2D_AF_sigmoid_KSI_40_KSF_3	23.1684	0.3037	0.8036	0.432
CNN2D_AF_sigmoid_KSI_50_KSF_3	20.141	0.2324	0.872	0.5707
CNN2D_AF_sigmoid_KSI_10_KSF_4	33.9582	0.4235	0.3704	0.2203
CNN2D_AF_sigmoid_KSI_20_KSF_4	25.5756	0.3047	0.7115	0.3078
CNN2D_AF_sigmoid_KSI_30_KSF_4	23.9408	0.2845	0.7848	0.3935
CNN2D_AF_sigmoid_KSI_40_KSF_4	21.7412	0.2627	0.8217	0.4998

Tabla 4.7 (cont). Resultados de las arquitecturas 2DCNN y ensamble completo.

Modelo	RMSE	MAPE	IOA	R2
CNN2D_AF_sigmoid_KSI_50_KSF_4	20.6608	0.2385	0.8622	0.5483
CNN2D_AF_sigmoid_KSI_10_KSF_5	33.7397	0.417	0.3648	0.2047
CNN2D_AF_sigmoid_KSI_20_KSF_5	34.0772	0.4271	0.3732	0.2289
CNN2D_AF_sigmoid_KSI_30_KSF_5	23.5693	0.2877	0.7729	0.4121
CNN2D_AF_sigmoid_KSI_40_KSF_5	20.6663	0.2468	0.8588	0.548
CNN2D_AF_sigmoid_KSI_50_KSF_5	20.7297	0.2593	0.8575	0.5452
CNN2D_AF_sigmoid_KSI_10_KSF_6	33.9702	0.4239	0.3707	0.2212
CNN2D_AF_sigmoid_KSI_20_KSF_6	23.8196	0.2976	0.7656	0.3996
CNN2D_AF_sigmoid_KSI_30_KSF_6	21.13	0.2611	0.8391	0.5275
CNN2D_AF_sigmoid_KSI_40_KSF_6	20.828	0.2517	0.8592	0.5409
CNN2D_AF_sigmoid_KSI_50_KSF_6	22.0727	0.2899	0.8472	0.4844
CNN2D_AF_sigmoid_KSI_10_KSF_7	22.6875	0.2521	0.7599	0.4553
CNN2D_AF_sigmoid_KSI_20_KSF_7	21.2051	0.2651	0.8398	0.5242
CNN2D_AF_sigmoid_KSI_30_KSF_7	20.9264	0.267	0.8524	0.5366
CNN2D_AF_sigmoid_KSI_40_KSF_7	21.7983	0.243	0.8542	0.4972
CNN2D_AF_sigmoid_KSI_50_KSF_7	21.8953	0.2627	0.8444	0.4927
CNN2D_AF_softplus_KSI_10_KSF_2	34.0733	0.4269	0.3731	0.2286
CNN2D_AF_softplus_KSI_20_KSF_2	22.8543	0.246	0.7421	0.4473
CNN2D_AF_softplus_KSI_30_KSF_2	22.9233	0.2474	0.7405	0.4439
CNN2D_AF_softplus_KSI_40_KSF_2	22.9213	0.2458	0.7392	0.444
CNN2D_AF_softplus_KSI_50_KSF_2	20.2219	0.2374	0.8648	0.5673
CNN2D_AF_softplus_KSI_10_KSF_3	23.1023	0.2572	0.7508	0.4352
CNN2D_AF_softplus_KSI_20_KSF_3	34.0583	0.4265	0.3728	0.2275
CNN2D_AF_softplus_KSI_30_KSF_3	22.8503	0.2503	0.7507	0.4474
CNN2D_AF_softplus_KSI_40_KSF_3	22.7934	0.2362	0.7393	0.4502
CNN2D_AF_softplus_KSI_50_KSF_3	20.8244	0.2334	0.8555	0.5411
CNN2D_AF_softplus_KSI_10_KSF_4	33.9812	0.4242	0.3709	0.222
CNN2D_AF_softplus_KSI_20_KSF_4	23.3581	0.2412	0.7076	0.4226
CNN2D_AF_softplus_KSI_30_KSF_4	20.7384	0.2615	0.844	0.5449
CNN2D_AF_softplus_KSI_40_KSF_4	20.9541	0.2768	0.8502	0.5354
CNN2D_AF_softplus_KSI_50_KSF_4	21.7232	0.25	0.8545	0.5006
CNN2D_AF_softplus_KSI_10_KSF_5	23.8021	0.2454	0.6887	0.4005
CNN2D_AF_softplus_KSI_20_KSF_5	23.041	0.2426	0.7283	0.4382
CNN2D_AF_softplus_KSI_30_KSF_5	19.9976	0.2336	0.8444	0.5768
CNN2D_AF_softplus_KSI_40_KSF_5	20.6199	0.2478	0.8504	0.5501
CNN2D_AF_softplus_KSI_50_KSF_5	20.5462	0.2469	0.8555	0.5533
CNN2D_AF_softplus_KSI_10_KSF_6	22.4072	0.236	0.7513	0.4687

Tabla 4.7 (cont). Resultados de las arquitecturas 2DCNN y ensamble completo.

Modelo	RMSE	MAPE	IOA	R2
CNN2D_AF_softplus_KSI_20_KSF_6	19.5143	0.2386	0.8751	0.597
CNN2D_AF_softplus_KSI_30_KSF_6	20.0261	0.2404	0.8614	0.5756
CNN2D_AF_softplus_KSI_40_KSF_6	20.1571	0.2384	0.8578	0.57
CNN2D_AF_softplus_KSI_50_KSF_6	21.1945	0.254	0.8542	0.5246
CNN2D_AF_softplus_KSI_10_KSF_7	20.3033	0.2595	0.8533	0.5638
CNN2D_AF_softplus_KSI_20_KSF_7	19.879	0.2298	0.8505	0.5818
CNN2D_AF_softplus_KSI_30_KSF_7	20.8116	0.2554	0.863	0.5416
CNN2D_AF_softplus_KSI_40_KSF_7	21.2755	0.2497	0.8544	0.521
CNN2D_AF_softplus_KSI_50_KSF_7	22.2947	0.2698	0.8265	0.474
Ensamble	19.8098	0.23	0.8419	0.5847

Esta arquitectura es la que tiene un mayor número de pruebas. El nombre de cada fila se define de la siguiente manera. El nombre comienza con CNN2D (CNN bidimensional), seguido de AF (función de activación) y la función de activación seleccionada. A continuación, se muestran KSI (dimensión del tamaño del núcleo de las entradas) y KSF (dimensión del tamaño del núcleo de las características) seguidos por el valor seleccionado para cada una de ellas respectivamente. Las tres mejores arquitecturas están en negrita.

El ensamble final de las arquitecturas 2DCNN es el que tiene mejores resultados de los ensambles de arquitectura única. Además, los resultados de este modelo y las arquitecturas seleccionadas se presentan en la Tabla 4.8.

Tabla 4.8. Ensamble final para arquitecturas 2DCNN.

Modelo	RMSE	MAPE	IOA	R2
CNN2D_AF_linear_KSI_30_KSF_2				
CNN2D_AF_linear_KSI_20_KSF_3				
CNN2D_AF_linear_KSI_40_KSF_3				
CNN2D_AF_linear_KSI_30_KSF_4				
CNN2D_AF_linear_KSI_50_KSF_6				
CNN2D_AF_relu_KSI_40_KSF_3	18.7697	0.2295	0.8805	0.6221
CNN2D_AF_relu_KSI_10_KSF_4				
CNN2D_AF_sigmoid_KSI_50_KSF_3				
CNN2D_AF_sigmoid_KSI_50_KSF_4				
CNN2D_AF_softplus_KSI_10_KSF_5				
CNN2D_AF_softplus_KSI_30_KSF_5				
CNN2D_AF_softplus_KSI_20_KSF_6				

Este modelo de ensamble presentó el mejor resultado de todos los modelos de ensamble de arquitectura única probados en este trabajo, aunque es una pequeña mejora en comparación con el 1DCNN con dos capas convolucionales.

Una vez probadas todas las variantes de la arquitectura, los cuatro ensambles resultantes se introdujeron en el modelo final del ensamble. El propósito de esto es determinar cuál es el mejor ensamble para esas arquitecturas, como se muestra en la tabla 4.9.

Tabla 4.9. Resultados de las cuatro arquitecturas de ensamble resultantes y ensamble completo.

Modelo	RMSE	MAPE	IOA	R2
CNN2D_AF_linear_KSI_20_KSF_3	19.2497	0.2314	0.8704	0.6079
CNN2D_AF_linear_KSI_30_KSF_2	19.6173	0.2467	0.8794	0.5927
CNN2D_AF_linear_KSI_30_KSF_4	19.7419	0.2453	0.8816	0.5876
CNN2D_AF_linear_KSI_40_KSF_3	20.4432	0.2549	0.8835	0.5577
CNN2D_AF_linear_KSI_50_KSF_6	19.2326	0.2317	0.8747	0.6086
CNN2D_AF_relu_KSI_10_KSF_4	21.59	0.2437	0.8275	0.5067
CNN2D_AF_relu_KSI_40_KSF_3	22.1329	0.239	0.8258	0.4816
CNN2D_AF_sigmoid_KSI_50_KSF_3	20.141	0.2324	0.872	0.5707
CNN2D_AF_sigmoid_KSI_50_KSF_4	20.6608	0.2385	0.8622	0.5483

Tabla 4.9 (cont). Resultados de las cuatro arquitecturas de ensamble resultantes y ensamble completo.

Modelo	RMSE	MAPE	IOA	R2
CNN2D_AF_softplus_KSI_10_KSF_5	23.8021	0.2454	0.6887	0.4005
CNN2D_AF_softplus_KSI_20_KSF_6	19.5143	0.2386	0.8751	0.597
CNN2D_AF_softplus_KSI_30_KSF_5	19.9976	0.2336	0.8444	0.5768
CNN_AF_linear_KS1_2_KS2_2	19.3009	0.233	0.8723	0.6058
CNN_AF_linear_KS1_2_KS2_3	19.4228	0.237	0.8752	0.6008
CNN_AF_linear_KS1_2_KS2_5	19.5858	0.2376	0.8604	0.594
CNN_AF_linear_KS1_2_KS2_6	19.3982	0.2319	0.8635	0.6018
CNN_AF_linear_KS1_4_KS2_2	19.6161	0.2356	0.8554	0.5928
CNN_AF_linear_KS1_5_KS2_2	19.6789	0.2363	0.8575	0.5902
CNN_AF_relu_KS1_3_KS2_3	22.2487	0.2544	0.8502	0.4762
CNN_AF_relu_KS1_6_KS2_2	22.7216	0.2937	0.8333	0.4537
CNN_AF_sigmoid_KS1_2_KS2_2	20.8244	0.2399	0.8792	0.5411
CNN_AF_sigmoid_KS1_2_KS2_3	20.2936	0.239	0.8781	0.5642
CNN_AF_sigmoid_KS1_2_KS2_4	20.3843	0.2407	0.8782	0.5603
CNN_AF_softplus_KS1_2_KS2_2	20.18	0.2463	0.8603	0.569
CNN_AF_softplus_KS1_2_KS2_3	21.4367	0.2583	0.8592	0.5137
CNN_AF_linear_KS_2	19.3736	0.2395	0.8782	0.6028
CNN_AF_linear_KS_5	19.4786	0.2353	0.8641	0.5985
CNN_AF_linear_KS_6	19.4022	0.2371	0.8695	0.6016
CNN_AF_sigmoid_KS_3	20.0372	0.2333	0.8749	0.5751
CNN_AF_softplus_KS_2	19.6784	0.2289	0.8743	0.5902
CNN_AF_softplus_KS_3	20.2479	0.24	0.8719	0.5661
MLP_AF_linear_DN_10	19.758	0.2369	0.8713	0.5869
MLP_AF_linear_DN_12	19.7803	0.2379	0.8732	0.586
MLP_AF_linear_DN_14	19.7654	0.2361	0.8632	0.5866
MLP_AF_linear_DN_16	19.7369	0.2353	0.8673	0.5878
MLP_AF_linear_DN_20	19.716	0.2352	0.8698	0.5886
MLP_AF_softplus_DN_14	20.0226	0.2375	0.8639	0.5757
Ensamble	18.7889	0.2218	0.8793	0.6264

La estructura del nombre de cada arquitectura puede consultarse en su correspondiente tabla de arquitectura individual. Las tres mejores arquitecturas están en negrita. Se observa que las tres mejores arquitecturas corresponden a una arquitectura de convolución.

Esta vez el ensamble contiene más arquitecturas que los anteriores presentados y se puede ver que la mayoría de las arquitecturas son de cualquier tipo de convolución, como se muestra en la tabla 4.10.

Tabla 4.10. Ensamble final para las cuatro arquitecturas de conjuntos resultantes.

Modelo	RMSE	MAPE	IOA	R2
CNN2D_AF_linear_KSI_40_KSF_3				
CNN2D_AF_relu_KSI_10_KSF_4				
CNN2D_AF_relu_KSI_40_KSF_3				
CNN2D_AF_sigmoid_KSI_50_KSF_3				
CNN2D_AF_softplus_KSI_10_KSF_5				
CNN2D_AF_softplus_KSI_20_KSF_6				
CNN2D_AF_softplus_KSI_30_KSF_5				
CNN_AF_linear_KS1_4_KS2_2	18.6875	0.2172	0.8807	0.6302
CNN_AF_relu_KS1_3_KS2_3				
CNN_AF_sigmoid_KS1_2_KS2_2				
CNN_AF_sigmoid_KS1_2_KS2_3				
CNN_AF_softplus_KS1_2_KS2_3				
CNN_AF_linear_KS_2				
MLP_AF_linear_DN_10				
MLP_AF_linear_DN_12				
MLP_AF_linear_DN_20				

Aproximadamente el 80% de las arquitecturas que intervienen en este conjunto son de tipo convolucional. Contiene una variedad de funciones de activación, pero la lineal es la más repetida.

En la Tabla 4.10 se muestran los modelos finales que devuelven el mejor RMSE de todas las arquitecturas probadas. Estos modelos y su ensamble fueron finalmente probados con el conjunto de datos completo usando todos los meses desde el 2000 hasta el 2018. Los resultados de esta última evaluación se presentan en la Tabla 4.11.

Tabla 4.11. Resultados de las arquitecturas finales y el conjunto.

Modelo	RMSE	MAPE	IOA	R2
MLP_AF_linear_DN_10	19.7493	0.2363	0.8688	0.5872
MLP_AF_linear_DN_12	19.7293	0.2358	0.8697	0.5881
MLP_AF_linear_DN_20	19.7881	0.2388	0.8693	0.5856
CNN_AF_linear_KS_2	19.3646	0.2377	0.8764	0.6032
CNN_AF_linear_KS1_4_KS2_2	19.3967	0.2373	0.874	0.6019
CNN_AF_relu_KS1_3_KS2_3	23.2674	0.2787	0.823	0.4271
CNN_AF_sigmoid_KS1_2_KS2_2	19.619	0.2397	0.8827	0.5927
CNN_AF_sigmoid_KS1_2_KS2_3	19.255	0.2276	0.8843	0.6076
CNN_AF_softplus_KS1_2_KS2_3	19.6816	0.2491	0.8712	0.5901
CNN2D_AF_linear_KSI_40_KSF_3	19.4736	0.2424	0.8751	0.5987
CNN2D_AF_relu_KSI_10_KSF_4	22.0373	0.2645	0.8333	0.4861
CNN2D_AF_relu_KSI_40_KSF_3	21.9845	0.2592	0.8321	0.4885
CNN2D_AF_sigmoid_KSI_50_KSF_3	19.8153	0.2524	0.8741	0.5845
CNN2D_AF_softplus_KSI_10_KSF_5	19.5534	0.2372	0.8617	0.5954
CNN2D_AF_softplus_KSI_20_KSF_6	19.5796	0.2269	0.8657	0.5943
CNN2D_AF_softplus_KSI_30_KSF_5	19.5914	0.227	0.8555	0.5938
Ensamble	18.9535	0.2266	0.8769	0.6198

La estructura del nombre de cada arquitectura puede consultarse en su correspondiente tabla de arquitectura individual. Las tres mejores arquitecturas están en negrita. Se observa que las tres mejores arquitecturas corresponden a una arquitectura de convolución.

Finalmente, en la Tabla 4.12 se presenta el conjunto resultante de este trabajo. Este es el que tiene un mejor rendimiento y es el modelo que representa mejor el conjunto completo de datos utilizado.

Tabla 4.12. Modelo final de ensamble.

Modelo	RMSE	MAPE	IOA	R2
MLP_AF_linear_DN_10				
MLP_AF_linear_DN_12				
CNN_AF_relu_KS1_3_KS2_3				
CNN_AF_sigmoid_KS1_2_KS2_2				
CNN_AF_sigmoid_KS1_2_KS2_3	14.9469	0.2433	0.8994	0.6789
CNN_AF_softplus_KS1_2_KS2_3				
CNN2D_AF_relu_KSI_10_KSF_4				
CNN2D_AF_sigmoid_KSI_50_KSF_3				
CNN2D_AF_softplus_KSI_20_KSF_6				
CNN2D_AF_softplus_KSI_30_KSF_5				

Este ensamble final contiene 10 arquitecturas, de las cuales el 80% son de tipo convolucional. Se observa que la RMSE disminuyó en un 20% aproximadamente para este conjunto, en comparación con los resultados presentados anteriormente.

Incluso cuando el RMSE disminuye de manera importante, un análisis del comportamiento de error entre las predicciones del conjunto y el valor real revela un hecho importante. El error aumenta debido a algunos valores atípicos que afectan a la medición. En la Figura 4.1 se muestra un diagrama de caja donde se muestra la distribución de errores entre las predicciones y los datos reales.

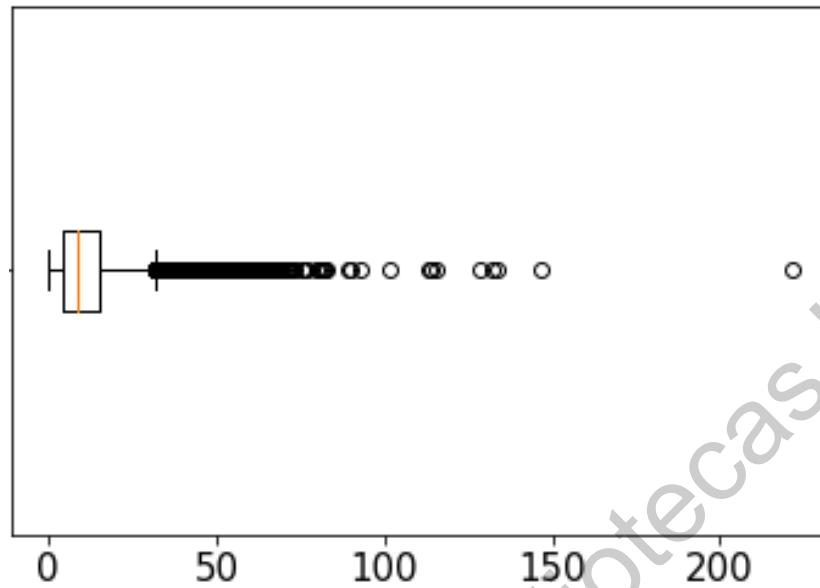


Figura 4.1. Distribución de errores entre predicciones y datos reales. En el eje X se presenta el error real de todo el conjunto de pruebas y su distribución.

Está claro que la mayoría de los valores de error se encuentran en valores más bajos. Tomando la media de los errores absolutos se obtiene un resultado de 10,9 $\mu\text{g}/\text{m}^3$ y la mediana es de 8,52 $\mu\text{g}/\text{m}^3$; luego, se observa que algunos de los valores atípicos son considerablemente grandes, por lo que aumenta la medición RMSE.

Al realizar las mismas evaluaciones con los conjuntos de cada arquitectura es evidente que este comportamiento está presente en todas ellas, como se muestra en la Figura 4.2.

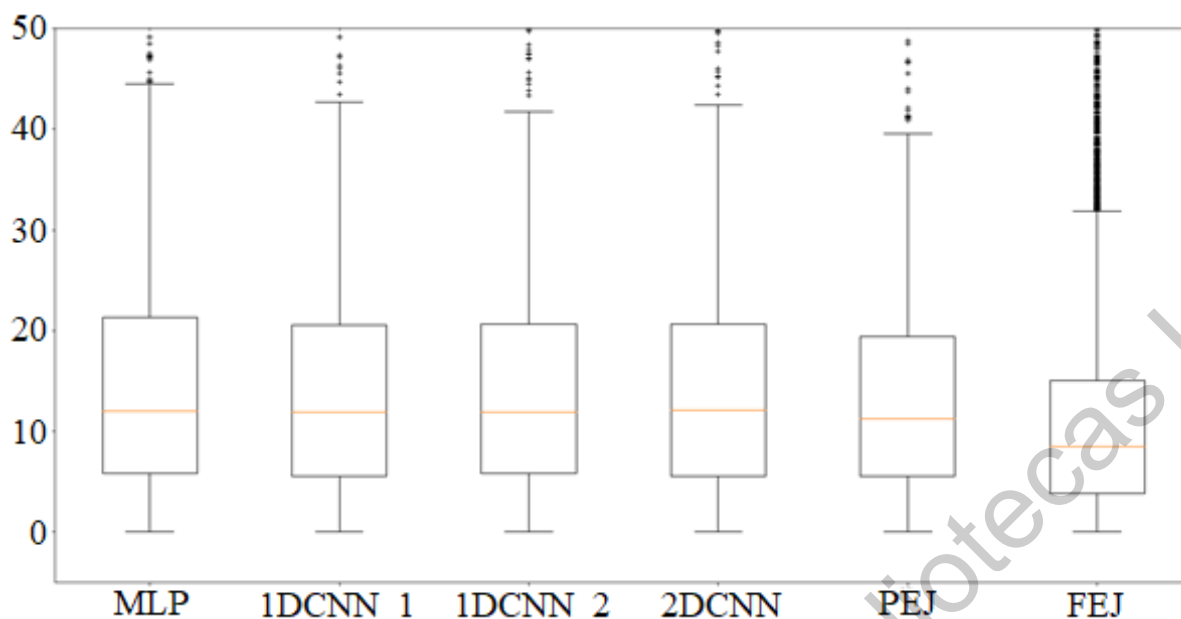


Figura 4.2. Distribución de errores en todas las arquitecturas. PEJ: Ensamble final con datos de Enero. FEJ: Ensamble Final con todo el conjunto de datos.

En la Figura 4.2, puede observarse que el orden de las arquitecturas se presenta por orden de aparición en esta obra. El primero es el perceptrón multicapa (MLP), el siguiente es el CNN unidimensional con una capa convolucional (1DCNN_1). El tercer modelo es el CNN unidimensional con dos capas convolucionales (1DCNN_2). El cuarto modelo es el CNN bidimensional (2DCNN). El quinto modelo es el conjunto parcial entrenado con los datos del mes de enero (PEJ). Finalmente, el sexto modelo es el conjunto final entrenado con el conjunto completo de datos (FEJ).

Se puede ver en la Figura 4.2 que el conjunto final entrenado y probado con el conjunto de datos completo tiene una media y mediana más baja que los otros modelos. Además, presenta un diagrama de caja más estrecho y más valores atípicos. Estos resultados implican que el modelo final tiene mayor nivel predictivo.

5. Discusión de Resultados

Analizando los resultados de cada arquitectura, se observa que en el caso de MLP las mejores arquitecturas tenían 10, 16 y 20 neuronas en la capa oculta. Todos ellos con la función de activación lineal. El mejor RMSE para MLP fue de 19.716 siendo la peor arquitectura de todas las probadas, lo que muestra que las arquitecturas 1DCNN y 2DCNN exhiben una disminución en el error que arquitecturas establecidas como la MLP. Esta arquitectura fue tomada para propósitos de comparación y los resultados de las otras tres arquitecturas se miden bajo este resultado.

En el caso de 1DCNN con una capa de convolución, los mejores resultados se obtuvieron de nuevo con la función de activación lineal y tamaños de núcleo de 2, 3 y 6. Siendo el tamaño del núcleo de 2 el que tiene el menor RMSE de 19.3736. Esto implica una mejora de aproximadamente el 1,8% frente al MLP.

De la misma manera, para 1DCNN con dos capas de convolución, la función de activación lineal fue una vez más la más adecuada. En el mismo sentido, los mejores tamaños de núcleo en ambas capas fueron 2, 3, 4 y 6. Esto es consistente con los resultados de 1DCNN con una capa de convolución. Para esta arquitectura el mejor RMSE fue el que tenía un tamaño de núcleo de 2 en ambas capas de convolución, con un valor de 19.3009. Esto representa una mejora del 2,1% frente al MLP.

Por último, para el 2DCNN, la función de activación lineal ha dado mejores resultados. Los mejores tamaños de núcleo para el eje de características fueron 3 y 6 continuando con los resultados de ambos 1DCNN. En el caso del tamaño del núcleo para el eje de horas anterior, los más adecuados fueron 20 y 50. El mejor RMSE para esta arquitectura fue de 19.2497, lo que representa una mejora aproximada de 2.4% sobre el MLP.

De esos datos se puede ver que la mejor función de activación para esta aplicación resultó ser la activación lineal. Además, se nota que los tamaños de *kernel* más pequeños tienden a ser mejores. En el modelo final se logran mejoras que están dadas, de manera consistente, por cada disminución del error en cada arquitectura de convolución.

Para el siguiente set, los conjuntos de arquitecturas individuales tuvieron un comportamiento similar. Para el MLP el RMSE fue de 19.7213, siendo un resultado peor que el modelo individual en 0.02%. En el caso del conjunto de 1DCNNN con una capa convolucional la RMSE fue de 19.0053, lo que representa una mejora del 3,6%. Para el 1DCNN con dos capas convolucionales el RMSE terminó en 18,9024, lo que implica una mejora del 4,1%. Finalmente, 2DCNNN obtuvo una puntuación de RMSE de 18,7697, lo que supone una mejora del 4,8%.

Después de los ensamblajes individuales, el ensamblaje con la mezcla de arquitecturas resultó en un 80% de arquitecturas convolucionales, con una puntuación de RMSE de 18,6875, lo que supone una mejora del 5,2%.

El último conjunto resultante se probó con el conjunto de datos completo, obteniendo un RMSE de 14.9469. Esta es una mejora más significativa del 24,2%. Una comparación gráfica de este resultado puede ser observado en la Figura 4.2, donde el *boxplot* del lado derecho representa el mejor modelo obtenido. Es decir, el que presenta un error promedio más bajo.

En la Figura 4.3 se observa un diagrama sencillo que representa el modelo final utilizado. En él se ven las arquitecturas usadas y la combinación de ellas mediante el ensamble bagging para la obtención final.

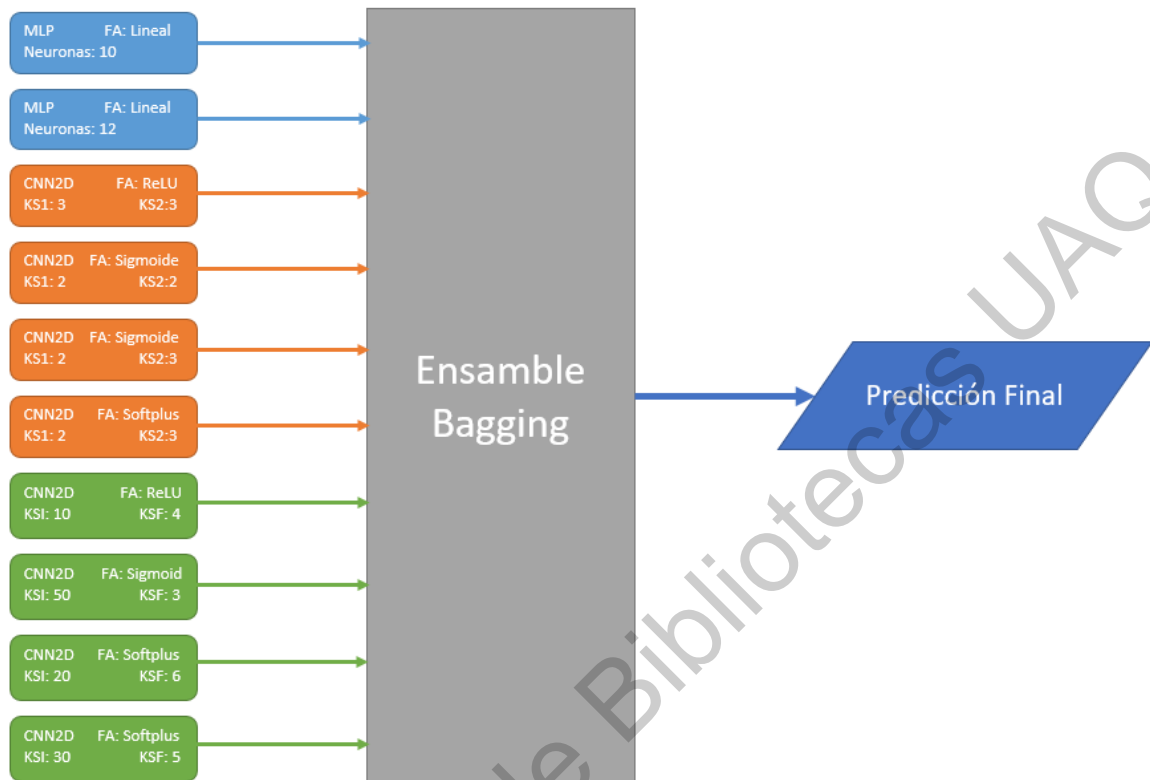


Figura 4.3. Diagrama de los componentes del modelo final para la predicción de PM10. [Creación propia]

Se tiene un conjunto pequeño, 10 arquitecturas, seleccionado de todo el conjunto inicial con las arquitecturas que presentaron un mejor desempeño para la predicción de PM10. Es importante recordar que todo este proceso se realizó de forma automática. La creación, entrenamiento, evaluación y selección de arquitecturas, así como la mejor combinación para el algoritmo de bagging, se realizó de manera automática por medio de la implementación aquí realizada.

Teniendo una visión general de todos los resultados, es evidente que los modelos de ensamble mejoran las capacidades de predicción en todos los casos, lo que nos dice que dichos algoritmos son de gran ayuda para la mejora de los modelos predictivos.

Por último, aunque la mejora de los resultados con base en MLP puede parecer insuficiente sólo con una CNN, se puede concluir que una combinación de las arquitecturas e hiperparámetros correctos con un método de ensamble representa una ventaja, para mejorar enormemente los resultados. Y considerando que los valores atípicos (Figura 4.2), de nuestro conjunto de predicciones final, comienzan a partir de valores de RMSE de aproximadamente 30. A diferencia de los demás modelos que muestran como valores atípicos valores de entre 40 a 45 de RMSE. Esto permite observar que, el modelo final cuenta con una menor cantidad de valores altos de RMSE, lo que representa una predicción más acertada en lo general.

Dirección General de Bibliotecas UAQ

6. Conclusiones y Trabajo Futuro

En este trabajo, se presenta un amplio estudio de las redes neuronales convolucionales y su comparación con el perceptrón multicapa. Además, se realiza un método de ensamble de *bagging* para mejorar la precisión del modelo. De los resultados presentados se puede observar que los modelos individuales con mayor precisión fueron los que tenían una capa de convolución. Aunque las diferencias son pequeñas en los modelos individuales, la acumulación de mejoras permite tener un mejor modelo cuando se combinan con un método de ensamble. En el modelo final obtenido, el 80% de los modelos individuales se basan en una arquitectura convolucional. Esto revela que las arquitecturas convolucionales pueden tener una mejor representación para esta aplicación.

Finalmente, el último método de ensamble de *bagging* presentó una mejora del 24.2% en relación a todos los mejores modelos individuales probados. Esta es una mejora significativa que nos permite saber que es una buena aproximación en la predicción de PM10.

Como trabajo futuro se sugiere basarse en los hiperparámetros obtenidos durante la experimentación de este trabajo y hacer un ajuste más extenso, junto con la consideración de otros hiperparámetros que pudieran tener un efecto en el resultado final. De igual manera, se recomienda incrementar la robustez de los resultados mediante la implementación de métodos como la validación cruzada. Dotando así al modelo de una prueba más certera de sus capacidades predictivas reales.

Adicionalmente se recomienda realizar una técnica de aumento de datos para generar una mayor base de datos que permita un mayor conjunto de entrenamiento, validación y prueba.

Referencias

Abderrahim, H.; Chellali, M. R.; Hamou, A. Forecasting PM 10 in Algiers : efficacy of multilayer perceptron networks. *ENVIRON SCI POLLUT RES* **2016**, 23(2), 1634-41.

Aceves-Fernández, M. A., Pedraza-Ortega, J. C., Sotomayor-Olmedo, A., Ramos-Arreguín, J. M., Vargas-Soto, J. E., et al.: Analysis of Key Features of NonLinear Behaviour Using Recurrence Quantification. Case Study: Urban Airborne Pollution at Mexico City. *Environmental Modeling & Assessment* 19(2), 139-152 (2014).

Antanasijević, D. Z., Pocajt, V. V., Povrenović, D. S., Ristić, M. Đ., & Perić-Grujić, A. A. (2013). PM10 emission forecasting using artificial neural networks and genetic algorithm input variable optimization. *Science of the Total Environment*, 443, 511-519.

Arora, R.; Basu, A.; Mianjy, P.; Mukherjee, A. Understanding deep neural networks with rectified linear units. *International Conference on Learning Representations, Vancouver Convention Center, Vancouver, BC, Canada, April 30 - May 3, 2018*; Bengio, Y., LeCun, Y.; ICLR 2018.

Asson, E. M.; Wang, Y.-J. Introduction to computation and learning in artificial neural networks. *European Journal of Operational Research* **1990**, 47(November 1989), 1–28.

Athira, V., Geetha, P., Vinayakumar, R., & Soman, K. P. (2018). DeepAirNet: Applying Recurrent Networks for Air Quality Prediction. *Procedia Computer Science*, 132, 1394–1403.

Beissinger, M. B. Deep generative networks for sequence prediction. Master of Science in Engineering, University of Pennsylvania, Philadelphia, Pennsylvania, U.S.A., 2017

Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J MACH LEARN RES* **2012**, 13(2012), 281–305.

Berk, R. A. An introduction to ensemble methods for data analysis. *SOCIOLOGICAL METHODS* **2006**, 34(3), 263-295.

Busia, A.; Jaitly, N. Next-step conditioned deep convolutional neural networks improve protein secondary structure prediction. Special Issue of Bioinformatics, Prague Congress Centre, Prague, Czech Republic, July 21-25 2017; Beerenwinkel, N.; Bromberg, Y.; Oxford University Press: Oxford, England, 2017.

Cabrera-Hernandez, M. C., Acevez-Fernandez, M. A., Ramos-Arreguín, J. M., Vargas-Soto, J. E. & Gorrostieta-Hurtado, E.: Parameters Influencing the Optimization Process in Airborne Particles PM10 Using a Neuro-Fuzzy Algorithm Optimized with Bacteria Foraging (BFOA). *International Journal of Intelligence Science* 9, 67-91 (2019).

Calidad del Aire. Available online: <http://www.aire.cdmx.gob.mx/default.php?opc=%27aKBhnmM=%27> (accessed on 25 June 2019).

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* **2002**, 287(3), 177-201.

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. *ENVIRON HEALTH PERSP* **2006**, 114 (6), 810-817.

Cortina-Januchs, M. G.; Quintanilla-Dominguez, J.; Vega-Corona, A.; Andina, D. Development of a model for forecasting of PM10 concentrations in Salamanca, Mexico. *ATMOS POLLUT RES* **2015**, 6(4), 626-634.

Dong, M.; Yang, D.; Kuang, Y.; He, D.; Erdal, S.; Kenski, D. Expert Systems with Applications PM 2.5 concentration prediction using hidden semi-Markov model-based times series data mining. *Expert Systems With Applications* **2010**, 36(5), 9046-9055.

Eigen, D.; Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. Proceedings of the 2015 IEEE International Conference on Computer Vision, CentroParque Convention Center, Santiago, Chile, 2015; Mortensen, E.; Fidler, S.; IEEE Computer Society: Washington, DC, U.S.A.

Elangasinghe, M. A., Singhal, N., Dirks, K. N., Salmond, J. A., & Samarasinghe, S. (2014). Complex time series analysis of PM10 and PM2.5 for a coastal site using artificial neural network modelling and k-means clustering. *Atmospheric Environment*, 94, 106-116.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

Graczyk, M.; Lasota, T.; Trawi, B.; Trawi, K. Comparison of bagging, boosting and stacking ensembles applied to real estate appraisal. *LECT NOTES ARTIF INT* 2010, 5991, 340–350.

He, K.; Shang, X.; Ren, S.; Sun, J. Delving deep into rectifiers : surpassing human-level performance on ImageNet classification. *CoRR* 2015, 1502.01852.

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; ... Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv:1408.5093v1 2014.

Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A Convolutional Neural Network for Modelling Sentences. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* 2014 (Volume 1: Long Papers), 655–665.

Khaniabadi, Y. O.; Goudarzi, G.; Daryanoosh, S. M.; Borgini, A.; Tittarelli, A.; De Marco, A. Exposure to PM10, NO2, and O3 and impacts on human health. *Environmental Science and Pollution Research* 2017, 24(3),

Kingma, D. P.; Ba, J. L. ADAM: A method for stochastic optimization. International Conference on Learning Representations, San Diego, California, U.S.A., May 7-9 2015; Bengio, Y., LeCun, Y.; ICLR 2015.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105). Kurt, A.; Gulbagci, B.; Karaca, F.; Alagha, O. An online air pollution forecasting system using neural networks. *ENVIRON INT* **2008**, *34*, 592–598.

Li, X.; Ding, Q.; Sun, J. Q. Remaining useful life estimation in prognostics using deep convolution neural networks. *RELIAB ENG SYST SAFE* **2018**, *172*, 1–11.

Liu, N.; Han, J.; Zhang, D.; Wen, S.; Liu, T. Predicting eye fixations using convolutional neural networks. *Proceedings of the 2015 IEEE International Conference on Computer Vision, CentroParque Convention Center, Santiago, Chile, 2015; Mortensen, E.; Fidler, S.; IEEE Computer Society: Washington, DC, U.S.A.*

Martinez-Zeron, E., Acevez-Fernandez, M. A., Gorrostieta-Hurtado, E., Sotomayor-Olmedo, A. & Ramos-Arreguin, J. M.: Method to Improve Airborne Pollution Forecasting by Using Ant Colony Optimization and Neuro-Fuzzy Algorithms. *International Journal of Intelligence Science* *4*, 81-90 (2014).

McCulloch, W. S.; Pitts, W. H. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics* **1943**, *5*, 115–133.

Michalski, R. S.; Carbonell, J. G.; Mitchell, T. M. *Machine Learning : an Artificial Intelligence Approach* **1983**. Springer Berlin Heidelberg.

Nagelkerke, N. J. D. A Note on a General Definition of the Coefficient of Determination. *BIOMETRIKA* **1991**, *78*(3), 691-692.

Ordieres, J. B.; Vergara, E. P.; Capuz, R. S.; Salazar, R. E. Neural network prediction model for fine particulate matter (PM 2.5) on the US - Mexico border in El Paso (Texas) and Ciudad Juarez (Chihuahua). *Environmental Modelling & Software* **2015**, *20*(2015), 547–559.

Pak, U., Kim, C., Ryu, U., Sok, K., & Pak, S. (2018). A hybrid model based on convolutional neural networks and long short-term memory for ozone concentration prediction.

Panigrahi, S.; Karali, Y.; Behera, H. S. Normalize time series and forecast using evolutionary neural network. *IJERT* **2013**, 2(9), 2518–2522.

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 R* **2011**, 18(2), 316–327.

Pope Iii, C. A.; Dockery, D. W.; Spengler, J. D.; Raizenne, M. E. Respiratory Health and PM 10 Pollution A Daily Time Series Analysis. *American Review of Respiratory Disease* **1991**, 144, 668–674.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

Rumelhart, D. E.; Hintont, G. E. Learning representations by back-propagating errors. *Nature* **1986** (323), 3–6.

Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks* **2015**, 61, 85–117.

Secretaría del Medio Ambiente. Disponible en línea: <https://www.sedema.cdmx.gob.mx/> (revisado el 25 Junio de 2019).

Shenouda, E. A. M. A. A quantitative comparison of different MLP activation functions in classification. *LECT NOTES COMPUT SC* **2006**, 3971, 849–857.

Simard, P. Y.; Steinkraus, D.; Platt, J. C. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. *ICDAR 2003*, 1–6.

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).

Sun, W.; Zhang, H.; Palazoglu, A.; Singh, A.; Zhang, W.; Liu, S. Science of the Total Environment 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* **2013**, *443*, 93–103.

Syarif, I.; Zaluska, E.; Prugel-bennett, A.; Wills, G. Application of bagging, boosting and stacking to intrusion detection. *LECT NOTES ARTIF INT* **2012**, *7376*, 593–602.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1492-1500).

Anexo I

Como resultado de éste trabajo se logró la aceptación de un artículo científico en el Congreso Mexicano de Inteligencia Artificial 2019. Se realizó una presentación oral del trabajo y el artículo fue publicado en la revista Research in Computing Science.

Certificado de participación en el Congreso



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 PM₁₀, 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 PM₁₀ ha sido estudiada en [1], donde se determina el porcentaje de cada elemento que compone las PM₁₀ siendo el carbono uno de los más grandes.

Por otra parte, el impacto de las altas concentraciones de PM₁₀ 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 PM₁₀ 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 PM₁₀, las PM_{2.5} que son partículas suspendidas con un diámetro menor a los 2.5 micrómetros y las menores a 0.1 micrómetros (PM_{0.1}).

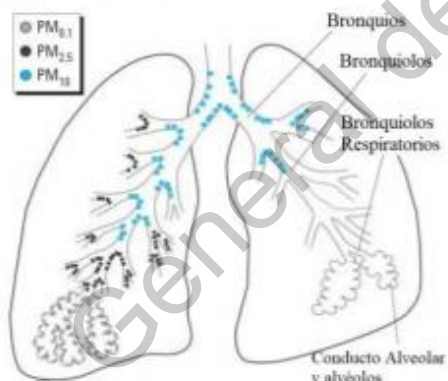


Fig. 1. Diagrama de puntos principales donde se concentran los subgrupos de PM en el sistema respiratorio [3].

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 (mW/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.

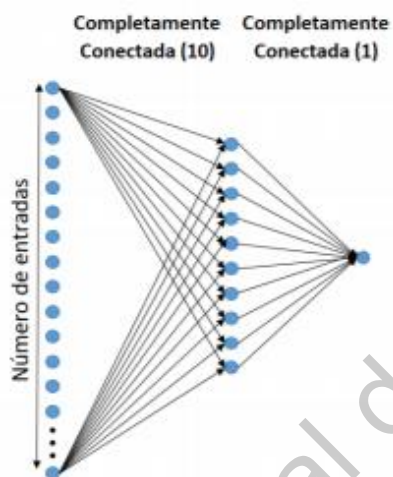


Fig. 3. Diagrama de MLP.

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.

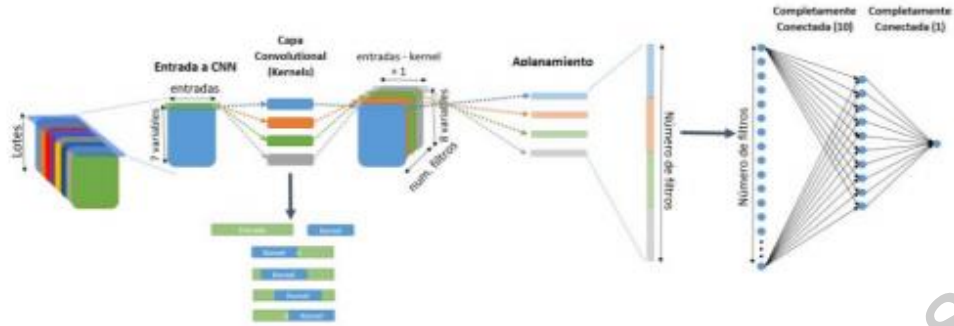


Fig. 4. Diagrama de CNN1D.

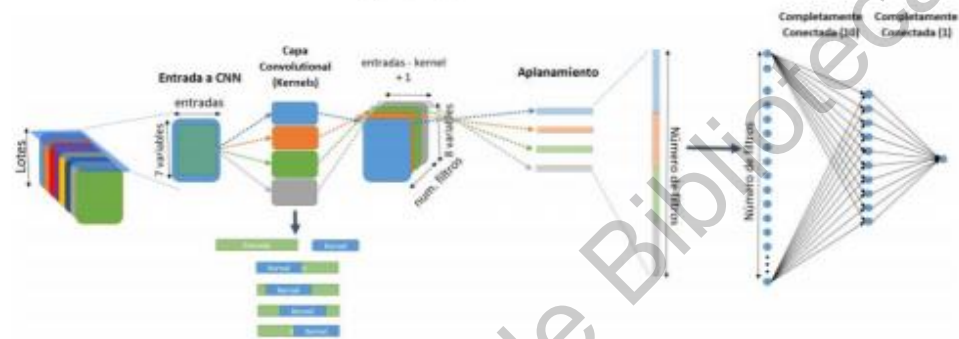


Fig. 5. Diagrama de la CNN2D.

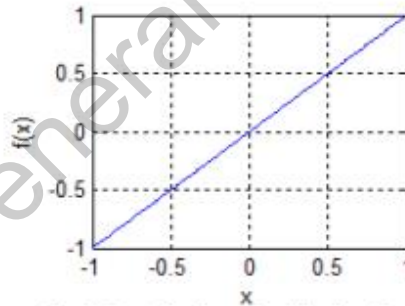


Fig. 6. Función de activación lineal.

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 número 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]:

$$f(x) = x. \tag{2}$$

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:

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

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:

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

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:

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

Adicionalmente se utiliza el Índice de Concordancia (IOA), obtenido mediante la ecuación (6). Se toma en cuenta la notación anterior y μ_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.

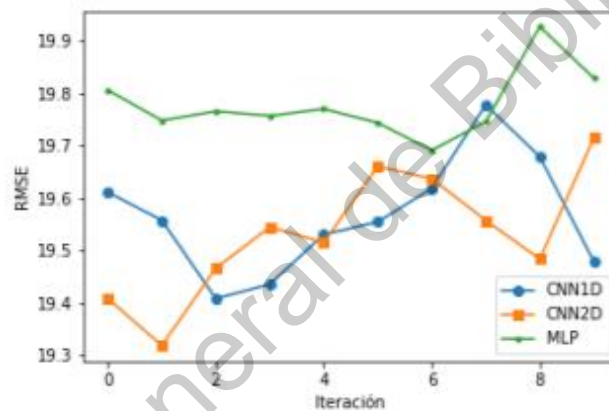


Fig. 7. Comparación de RMSE en 10 iteraciones con cada arquitectura.

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.

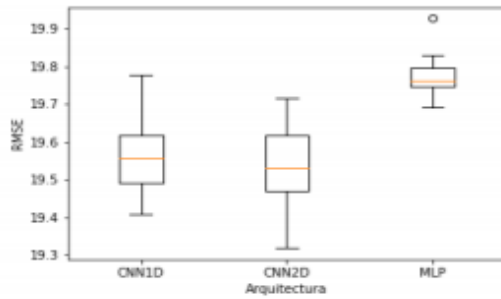


Fig. 8. Comparación de RMSE mediante diagrama de cajas.

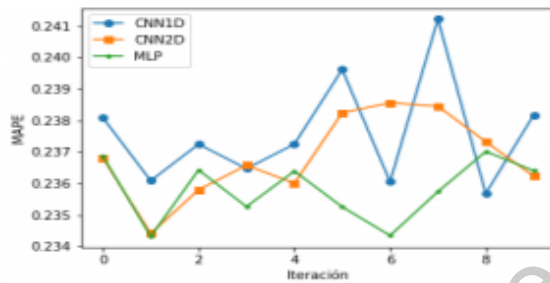


Fig. 9 Comparación de MAPE para las 3 arquitecturas.

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.

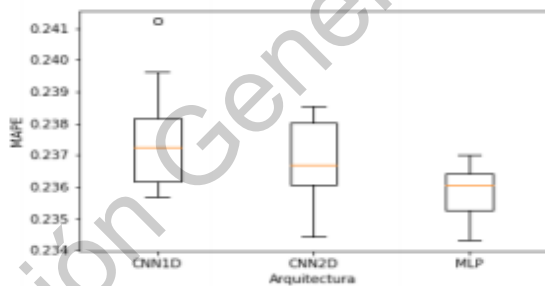


Fig. 10. Diagrama de cajas de MAPE.

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.

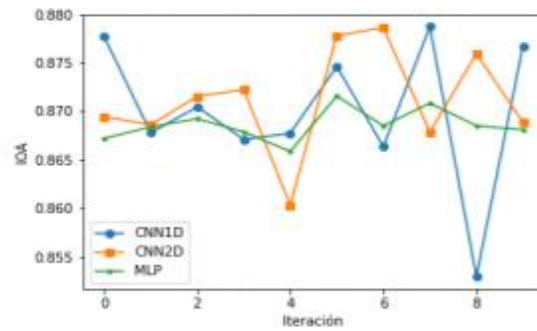


Fig. 11. Gráfica de comparación de IOA.

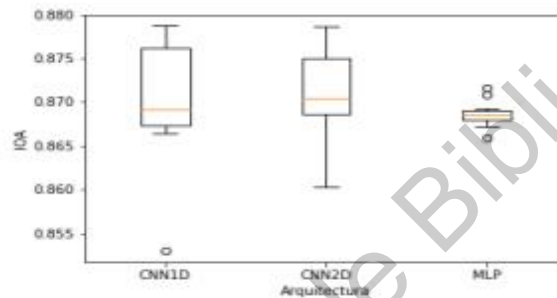


Fig. 12. Diagrama de cajas de IOA.

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
	MAPE	0.2358
	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 PM 10 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)

Anexo II

Obtención del certificado de manejo del idioma Inglés como requisito de egreso.



 UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE LENGUAS Y LETRAS 

Campus Aeropuerto, 06 de diciembre de 2019.
FLL/SA/HCA/2019/718

C. Domínguez Guevara, Ricardo
Exp. 275310
Facultad de Ingeniería
Presente.

Me permito comunicar a usted que en Sesión Ordinaria del H. Consejo Académico de la Facultad de Lenguas y Letras, celebrada el **06 de diciembre de 2019**, se acordó **Aprobar** su solicitud de **convalidación** de la Constancia del Examen de Manejo de la Lengua (8-), expedida por el CELYC, por el **Examen de Comprensión de Textos en Inglés**.

Sin más por el momento, me es grato quedar a sus órdenes para cualquier aclaración.

Atentamente,
"Enlazar Culturas por la Palabra"


Dra. Ma. de Lourdes Rico Cruz
Secretaría Académica


SECRETARÍA ACADÉMICA

C.c.: Expediente.
Archivo.
MISC/mtp

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
Campus Aeropuerto, Anillo Vial Fray Junipero Serra S/N, Querétaro, Qro.
C.P. 76140, Tel. 01 (442) 192 12 00 Dirección Ext. 61010,
Secretaría Administrativa 61300, Posgrado 61140, Licenciatura 61070,
Centro de Lenguas 61050, Secretaría Académica 61100 v Planeación 61110

SOMOS UAQ
EDUCAR CRECER CONSOLIDAR



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE LENGUAS Y LETRAS

A QUIEN CORRESPONDA:

La que suscribe, Directora Provisional de la Facultad de
Lenguas y Letras, hace **C O N S T A R** que

DOMINGUEZ GUEVARA RICARDO

Presentó el **Examen de Manejo de la Lengua** efectuado
el día siete de junio de dos mil dieciocho, en el cual obtuvo
la siguiente calificación:

8-

Se extiende la presente a petición de la parte interesada,
para los fines escolares y legales que le convengan, en el
Campus Aeropuerto de la Universidad Autónoma de
Querétaro, el día catorce de junio de dos mil dieciocho.



Atentamente
"Enlazar Culturas por la Palabra"

LIC. LAURA PÉREZ TÉLLEZ



LPT/evm*CL*FLL-C.-352



Campus Aeropuerto, Anillo Vial Fray Junipero Serra 5/N, Querétaro, Qro. C.P. 76140
Tel. 01 (442) 192 12 00 Dirección Ext. 61010, Secretaría Administrativa 61300, Posgrado 61140,
Licenciatura 61070, Centro de Lenguas 61050, Secretaría Académica 61100 y Planeación 61110