

UNIVERSIDAD TÉCNICA DE AMBATO



FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL MAESTRÍA EN MATEMÁTICA APLICADA

TEMA: “Pronóstico de precipitación en el centro de Tungurahua aplicando aprendizaje estadístico con redes neuronales artificiales”

Trabajo de titulación previo a la obtención del Grado Académico de Magister en Matemática Aplicada

Modalidad de Titulación: Proyecto de desarrollo

Autor: Ing. Juan Luis López Parra

Director(a): Ing. Marco Enrique Benalcázar Palacios, Ph.D.

Ambato – Ecuador

2023

APROBACIÓN DEL TRABAJO DE TITULACIÓN

A la Unidad Académica de Titulación de la Facultad de Ingeniería en Sistemas,
Electrónica e Industrial.

El Tribunal receptor de la Defensa del Trabajo de Titulación presidido por Ingeniera Elsa Pilar Urrutia Urrutia Magister, e integrado por los señores: Ingeniero Héctor Alberto Luzuriaga Jaramillo Magister y el Ingeniero Víctor Santiago Manzano Villafuerte Magister, designados por la Unidad Académica de Titulación de la Universidad Técnica de Ambato, para receptor el Trabajo de Titulación con el tema: **“PRONÓSTICO DE PRECIPITACIÓN EN EL CENTRO DE TUNGURAHUA APLICANDO APRENDIZAJE ESTADÍSTICO CON REDES NEURONALES ARTIFICIALES”**, elaborado y presentado por el señor, Ingeniero Juan Luis López Parra, para optar por el Grado Académico de Magister en Matemática Aplicada; una vez escuchada la defensa oral del Trabajo de Titulación el Tribunal aprueba y remite el trabajo para uso y custodia en las bibliotecas de la Universidad Técnica de Ambato.

Ing. Elsa Pilar Urrutia Urrutia Mg.
Presidente y Miembro del Tribunal de Defensa

Ing. Héctor Alberto Luzuriaga Jaramillo Mg.
Miembro del Tribunal de Defensa

Ing. Víctor Santiago Manzano Villafuerte Mg.
Miembro del Tribunal de Defensa

AUTORÍA DEL TRABAJO DE INVESTIGACIÓN

La responsabilidad de las opiniones, comentarios y críticas emitidas en el trabajo de titulación presentado con el tema: “Pronóstico de precipitación en el centro de Tungurahua aplicando aprendizaje estadístico con redes neuronales artificiales”, le corresponde exclusivamente a: Ingeniero Juan Luis López Parra, autor bajo la dirección del Ing. Marco Enrique Benalcázar Palacios, Ph.D., Director del Trabajo de Titulación; y el patrimonio intelectual a la Universidad Técnica de Ambato.

Ing. Juan Luis López Parra
AUTOR

Ing. Marco Enrique Benalcázar Palacios, Ph.D.
DIRECTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que el Trabajo de Titulación, sirva como un documento disponible para su lectura, consulta y procesos de investigación, según las normas de la Institución. Cedo los Derechos de mi trabajo, con fines de difusión pública, además apruebo la reproducción de este, dentro de las regulaciones de la Universidad Técnica de Ambato.

Ing. Juan Luis López Parra
C.I. 1804269320

ÍNDICE GENERAL DE CONTENIDOS

PORTADA.....	i
APROBACIÓN DEL TRABAJO DE TITULACIÓN	ii
AUTORÍA DEL TRABAJO DE INVESTIGACIÓN.....	iii
DERECHOS DE AUTOR	iv
ÍNDICE GENERAL DE CONTENIDOS.....	v
ÍNDICE DE TABLAS	viii
ÍNDICE DE FIGURAS.....	x
AGRADECIMIENTO	xii
DEDICATORIA	xiii
RESUMEN EJECUTIVO.....	xiv
CAPÍTULO I	1
EL PROBLEMA DE INVESTIGACIÓN.....	1
1.1. Introducción	1
1.2. Justificación	1
1.3. Objetivos	3
1.3.1. General.....	3
1.3.2. Específicos	3
CAPITULO II.....	4
ANTECEDENTES INVESTIGATIVOS	4
2.1. Redes neuronales	6
2.2. Redes neuronales recurrentes.....	7
2.3. Aplicación de redes neuronales recurrentes.....	8
2.4. Software Matlab.....	9
2.4.1. Deep network designer.....	9
2.4.2. Entrenar redes con Deep Network Designer	10
2.4.3. Datos de validación o Validation data	12
2.4.4. Datos de entrenamiento o Data Train.....	13
2.4.5. Data de testeo o Data test	14
2.4.6. Función de activación en el campo de redes neuronales.....	14
2.4.7. Función de activación Tangente hiperbólica (Tanh).....	15
2.4.8. Función de activación Rectified Lineal Unit (Relu)	16
2.4.9. LSTM (Long Short-Term Memory)	16
2.4.10. Regresión lineal en el campo de redes neuronales.....	17

2.4.11. Normalización de datos.....	17
2.4.12. Error RMSE	17
CAPITULO III.....	18
MARCO METODOLÓGICO.....	18
3.1. Ubicación.....	18
3.2. Equipos y materiales	18
3.3. Tipo de investigación.....	19
3.4. Recolección de información.....	19
3.5. Hipótesis	19
3.6. Estructura de la Información DATASET.....	19
3.7. Variables	20
3.8. Preprocesamiento de datos.....	21
3.9. Arquitectura base	21
3.10. Método de entrenamiento de la red neuronal.....	23
3.11. Experimentación y validación de la arquitectura	23
3.12. Desestandarización de datos	24
3.13. Validación del modelo	25
3.13.1. Graficación y cálculo de error.....	25
3.13.2. Variaciones y mejoras a los modelos	25
3.13.3. TESTEO.....	26
CAPITULO IV.....	27
RESULTADOS.....	27
4.1. Modelo 1	28
4.1.1 Variación 1.....	28
4.1.2 Variación 2.....	29
4.2. MODELO 2	30
4.2.1. Variación 1.....	31
4.2.2 Variación 2.....	31
4.2.3. Variación 3.....	32
4.2.4 Variación 4.....	33
4.3. MODELO 3	34
4.3.1. Variación 1.....	35
4.3.2. Variación 2.....	36
4.3.3 Variación 3.....	36
4.4. MODELO 4	38

4.4.1 Variación 1	38
4.4.2. Variación 2	39
4.4.3. Variación 3	40
4.4.4. Variación 4	40
4.5. MODELO 5	42
4.5.1. Variación 1	42
4.5.2. Variación 2	43
4.5.3. Variación 3	44
4.5.4. Variación 4	45
4.6. MODELO FINAL.....	47
CAPITULO V	50
CONCLUSIONES Y RECOMENDACIONES.....	50
5.1. Conclusiones	50
5.2. Recomendaciones	51
BIBLIOGRAFÍA	52
ANEXOS	57
Anexo 1 RMSE obtenido en experimentación.....	57
Anexo 2 Código de MatLab utilizado en el proyecto	66

ÍNDICE DE TABLAS

Tabla 1 Ecuación de la función de activación Tangente hiperbólica (Tanh)	15
Tabla 2 Ecuación de la Función de activación Rectified Lineal Unit (Relu).....	16
Tabla 3 División del DATA SET.....	20
Tabla 4 Parámetros definidos para método de entrenamiento	23
Tabla 5 Arquitectura del Modelo 1	28
Tabla 6 Variación 1 aplicada a la arquitectura del Modelo 1.....	28
Tabla 7 Variación 2 aplicada a la arquitectura del Modelo 1.....	29
Tabla 8 Arquitectura del Modelo 2	30
Tabla 9 Variación 1 aplicada a la arquitectura del Modelo 2.....	31
Tabla 10 Variación 2 aplicada a la arquitectura del Modelo 2.....	31
Tabla 11 Variación 3 aplicada a la arquitectura del Modelo 2.....	32
Tabla 12 Variación 4 aplicada a la arquitectura del Modelo 2.....	33
Tabla 13 Arquitectura del Modelo 3	34
Tabla 14 Variación 1 aplicada a la arquitectura del Modelo 3.....	35
Tabla 15 Variación 2 aplicada a la arquitectura del Modelo 3.....	36
Tabla 16 Variación 3 aplicada a la arquitectura del Modelo 3.....	36
Tabla 17 Arquitectura del Modelo 4	38
Tabla 18 Variación 1 aplicada a la arquitectura del Modelo 4.....	38
Tabla 19 Variación 2 aplicada a la arquitectura del Modelo 4.....	39
Tabla 20 Variación 3 aplicada a la arquitectura del Modelo 4.....	40
Tabla 21 Variación 4 aplicada a la arquitectura del Modelo 4.....	40
Tabla 22 Arquitectura del Modelo 5	42
Tabla 23 Variación 1 aplicada a la arquitectura del Modelo 5.....	42
Tabla 24 Variación 2 aplicada a la arquitectura del Modelo 5.....	43
Tabla 25 Variación 3 aplicada a la arquitectura del Modelo 5.....	44
Tabla 26 Variación 4 aplicada a la arquitectura del Modelo 5.....	45
Tabla 27 Arquitectura del Modelo Final.....	47
Tabla 28 Parámetros de entrenamiento del Modelo Final.....	47
Tabla 29 RMSE en experimentación con FA tanh.....	57
Tabla 30 RMSE en experimentación con FA relu	57
Tabla 31 RMSE en experimentación con 2 Bloques CC+FA+D, FA: Tanh y Relu.....	58
Tabla 32 RMSE en experimentación con 2 Bloques CC+FA+D, FA: Relu y Tanh.....	59
Tabla 33 RMSE en experimentación con 2 Bloques CC+FA+D, FA: Relu y Relu.....	60
Tabla 34 RMSE en experimentación con 2 Bloques CC+FA+D, FA: Tanh y Tanh	61
Tabla 35 RMSE en experimentación con 3 Bloques CC+FA+D, FA: Relu, Relu y Relu	62

Tabla 36	RMSE en experimentación con 3 Bloques CC+FA+D, FA: Relu, Relu y Tanh ...	62
Tabla 37	RMSE en experimentación con 3 Bloques CC+FA+D, FA: Relu, Tanh y Relu ...	63
Tabla 38	RMSE en experimentación con 3 Bloques CC+FA+D, FA: Relu, Tanh y Tanh...	63
Tabla 39	RMSE en experimentación con 3 Bloques CC+FA+D, FA: Tanh, Relu y Relu ...	64
Tabla 40	RMSE en experimentación con 3 Bloques CC+FA+D, FA: Tanh, Relu y Tanh...	64
Tabla 41	RMSE en experimentación con 3 Bloques CC+FA+D, FA: Tanh, Tanh y Relu...	65
Tabla 42	RMSE en experimentación con 3 Bloques CC+FA+D, FA: Tanh, Tanh y Tanh..	65

ÍNDICE DE FIGURAS

Figura 1 Red neuronal multicapa.....	7
Figura 2 Estructura de una red neuronal recurrente.....	8
Figura 3 Deep Network Designer	10
Figura 4 Proceso de entrenamiento.....	12
Figura 5 Datos de validación	13
Figura 6 Modelo neurona artificial	15
Figura 7 Estación Meteorológica de Querochaca	18
Figura 8 Datos y subprocesos de organización.....	20
Figura 9 Capas presentes en el Bloque CC + FA + D.....	22
Figura 10 Arquitectura general de la red neuronal	23
Figura 11 Capas de experimentación.....	24
Figura 12 Variabilidad en el Bloque CC + FA + D	24
Figura 13 Comparación de Precipitación.....	27
Figura 14 Pronóstico generado por el modelo 1 en comparación al original, y Diferencia entre valor pronóstico y valor real	28
Figura 15 Pronóstico generado por la variación 1 del modelo 1 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	29
Figura 16 Pronóstico generado por la variación 2 del modelo 1 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	29
Figura 17 Pronóstico generado por el modelo 2 en comparación al original, y Diferencia entre valor pronóstico y valor real	30
Figura 18 Pronóstico generado por la variación 1 del modelo 2 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	31
Figura 19 Pronóstico generado por la variación 2 del modelo 2 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	32
Figura 20 Pronóstico generado por la variación 3 del modelo 2 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	33
Figura 21 Pronóstico generado por la variación 4 del modelo 2 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	34
Figura 22 Pronóstico generado por el modelo 3 en comparación al original, y Diferencia entre valor pronóstico y valor real	35

Figura 23 Pronóstico generado por la variación 1 del modelo 3 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	35
Figura 24 Pronóstico generado por la variación 2 del modelo 3 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	36
Figura 25 Pronóstico generado por la variación 3 del modelo 3 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	37
Figura 26 Pronóstico generado por el modelo 4 en comparación al original, y Diferencia entre valor pronóstico y valor real	38
Figura 27 Pronóstico generado por la variación 1 del modelo 4 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	39
Figura 28 Pronóstico generado por la variación 2 del modelo 4 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	39
Figura 29 Pronóstico generado por la variación 3 del modelo 4 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	40
Figura 30 Pronóstico generado por la variación 4 del modelo 4 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	41
Figura 31 Pronóstico generado por el modelo 5 en comparación al original, y Diferencia entre valor pronóstico y valor real	42
Figura 32 Pronóstico generado por la variación 1 del modelo 5 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	43
Figura 33 Pronóstico generado por la variación 2 del modelo 5 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	44
Figura 34 Pronóstico generado por la variación 3 del modelo 5 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	45
Figura 35 Pronóstico generado por la variación 4 del modelo 5 en comparación al original, y Diferencia entre valor pronóstico y valor real.....	46
Figura 36 Pronóstico generado por el modelo Final en comparación al comportamiento original, y Diferencia entre valor pronóstico y valor real	48

AGRADECIMIENTO

Primeramente, a Dios, quien es mi todo. Él me da fuerzas cada día, me llena con su amor, me ayuda en cada paso y en cada proyecto, me da fuerzas y me guía. Dios es mi razón para vivir, existir, para cantar y soñar, le agradezco por tener el privilegio ser un hijo suyo.

Agradezco, a mi esposa Karen Pintado que me ha acompañado en cada logro importante y me ha motivado para seguir adelante en mi preparación académica y a cumplir metas. Agradezco también a mi madre Giovanna, a mi hermano Pablo, a mi abuelita Fabiola, que han estado presentes en el transcurso de mi vida estudiantil y son parte importante de mi vida.

Le doy muchas gracias al PhD. Marco E. Benalcázar, quién me ha apoyado en el proyecto de tesis, compartiendo su conocimiento y ayudándome a progresar en mi preparación, es un excelente profesional.

Juan Luis

DEDICATORIA

Dedico este proyecto a Dios, mi padre celestial, quien me ha guiado en cada paso que he dado en la maestría, y que me sostiene y protege cada día.

A mi esposa Karen Pintado, por motivarme a no rendirme y acompañarme siendo mi ayuda idónea. A mi madre Giovanna Parra, que con su ejemplo de lucha incansable me ha inspirado para ser cada día mejor y entregar todo proyecto en manos de Dios.

Juan Luis

UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL
MAESTRÍA EN MATEMÁTICA APLICADA

TEMA:

Pronóstico de precipitación en el centro de Tungurahua aplicando aprendizaje estadístico con redes neuronales artificiales

AUTOR: Ing. Juan Luis López Parra

DIRECTOR(A): Ing. Marco Enrique Benalcázar Palacios Ph.D.

LÍNEA DE INVESTIGACIÓN: Energía, Desarrollo Sostenible y Gestión de Recursos Naturales

FECHA: 04 de diciembre de 2022

RESUMEN EJECUTIVO

Este trabajo presenta el análisis de aplicación de redes neuronales artificiales recurrentes para el pronóstico de precipitación en la estación meteorológica de Querochaca, provincia de Tungurahua. Los datos analizados pertenecen a la lluvia del periodo de 2015 al 2021. El modelo propuesto consiste en una red neuronal artificial recurrente de 5 capas. La primera capa es la celda de memoria LSTM, que se destaca por tomar en cuenta la secuencialidad de los datos que son objeto de análisis. Además, se utilizaron 4 capas completamente conectadas, donde la primera trabaja con función de activación tanh y 100 neuronas. Las dos capas que siguen a continuación tienen función de activación relu con 500 neuronas cada una. Estas tres capas mencionadas, trabajaron con factor dropout de 0,02. La capa de salida tiene 1 neurona con función de activación purelin. La arquitectura general de la red neuronal consiste en una primera capa con celda de memoria LSTM, seguida de tres capas ocultas y finalmente la capa de salida. La red neuronal ha sido configurada para desarrollar regresión lineal. Se realizaron pruebas con 10, 50, 100, 500 y 1000 neuronas respectivamente en la segunda, tercera y cuarta capa. Además, se trabajó con 10 y 100 unidades ocultas en la celda LSTM. Se seleccionó la mejor arquitectura en base a dos indicadores de rendimiento que son el error RMSE y la gráfica de pronóstico. Los hiperparámetros de entrenamiento que más influyen sobre el pronóstico son el valor máximo de épocas y la tasa de aprendizaje inicial. En este proyecto se obtuvieron los mejores resultados utilizando pocas unidades ocultas en la celda LSTM. El mejor modelo construido en

el presente trabajo presenta un error 4,4535 con una gráfica de pronóstico muy similar al comportamiento real.

DESCRIPTORES: Redes neuronales artificiales, Pronóstico, Precipitación, LSTM, Función de activación, Tungurahua.

CAPÍTULO I

EL PROBLEMA DE INVESTIGACIÓN

1.1. Introducción

En el presente proyecto se utiliza machine learning, a través de redes neuronales artificiales para realizar el pronóstico de precipitación en Querochacha provincia de Tungurahua. Para la aplicación de redes neuronales se utilizó el historial de precipitaciones de 7 años, desde 2015 hasta 2021. Los datos utilizados se obtuvieron de la red pública del Instituto Nacional de Meteorología e Hidrología, generados en la estación meteorológica de Querochacha mediante el software Matlab se realizó el procesamiento estadístico y la aplicación de redes neuronales artificiales. En el desarrollo del tema se experimentó con la cantidad de capas y neuronas, para obtener la arquitectura ideal que permita predecir la cantidad de lluvia en el lugar analizado.

Con base en los resultados, se puede utilizar el recurso informático desarrollado para determinar la cantidad de lluvia en las diferentes temporadas del año en Querochacha. La herramienta desarrollada puede ser de gran utilidad en la agricultura, en infraestructura civil, y en utilización de agua potable. Mediante el uso de la red neuronal planteada, se pueden justificar la aplicación de políticas de regulación de uso del agua, así como la construcción de obras hidráulicas de control de flujo de agua.

1.2. Justificación

Sudamérica es una región con una amplia diversidad de pisos climáticos, debido a la presencia de estructuras topográficas determinantes. La cordillera de los Andes y valles extensos con superficies acuáticas importantes como el Amazonas, además de la presencia predominante de la selva tropical amazónica generan variabilidad climática (Reboita et al., 2010). La región tiene un alto índice de precipitaciones en comparación con otras zonas del mundo, sin embargo, la distribución no es homogénea. En algunas zonas de la Sudamérica se da un fenómeno de estrés hídrico, además no existe una distribución estructurada y regulada del flujo hídrico (Zipparro y Morellato, 2004).

Las lluvias tempestuosas, la escorrentía alta de los ríos y las inundaciones se deben a las variaciones pronunciadas de temperatura en países como el nuestro, Ecuador. La

serranía ecuatoriana posee un sinnúmero de zonas climáticas con características propias, donde se dan distintos ecosistemas con sus características propias. La precipitación en los Andes ecuatorianos, se debe en su mayor parte al choque de masas de aire provenientes del mar y de las cadenas montañosas (Vuille et al., 2000). Es imprescindible contar con técnicas que permitan tener una idea del comportamiento de la lluvia para gestionar de forma adecuada el agua.

En base a lo mencionado, es necesario desarrollar herramientas que permitan tener un criterio claro en cuanto al comportamiento de fenómenos como la precipitación. Existen técnicas informáticas que se están utilizando en la actualidad para el análisis del comportamiento climático. La técnica de Machine Learning es un método muy útil para el ajuste no lineal. Esta técnica ayuda en la adquisición de valores en base a la predicción de variables independientes. Con una base de datos adecuada, desarrolla algoritmos apropiados y tiene la capacidad de aprender correlaciones numéricas en el campo climatológico (Li et al., 2017). Esta técnica ha tomado significativa relevancia en el campo energético y climatológico, sobre todo en el área de estimación de parámetros meteorológicos en base de datos estructurados, semiestructurados y no estructurados (Ordoñez-Palacios et al., 2020).

Con base en datos meteorológicos pasados, normalmente se llevan a cabo estudios para encontrar patrones que puedan relacionarse con eventos futuros. Las relaciones encontradas en los estudios y las condiciones climáticas de un contexto temporal presente predicen condiciones futuras. Los métodos estadísticos son de gran importancia, particularmente para la predicción meteorológica a largo plazo (Wang et al., 2020). Una base de datos sólida con datos meteorológicos específicos se convierte en el soporte del estudio a realizarse, debido a que contiene patrones que pueden vincularse con escenarios futuros climatológicos. Los métodos estadísticos poseen una relevancia alta, debido a que aporta significativamente para la predicción. Las redes neuronales artificiales utilizan eficientemente el ámbito estadístico en la predicción del comportamiento de parámetros meteorológicos (Makridakis et al., 2018). La finalidad de utilizar las predicciones pueden ser para el área agrícola y estructural, así como para la prevención de posibles fenómenos atmosféricos de gran magnitud.

Tungurahua es una provincia que destaca por su actividad agrícola, por lo que es importante conocer la cantidad de agua que potencialmente se podría usar. Es

necesario generar planes de riego, y establecer lineamientos de uso de agua en la provincia. El propósito de este proyecto es utilizar la técnica de machine learning, para pronosticar la precipitación a través del uso de redes neuronales recurrentes; y de esta forma poder establecer herramientas para la gestión hídrica a nivel de la región.

1.3. Objetivos

1.3.1. General

- Desarrollar una aplicación de machine learning utilizando redes neuronales para el pronóstico de precipitación.

1.3.2. Específicos

- Obtener una base de datos de parámetros meteorológicos relacionados a la precipitación, de al menos 5 años, y colocarlos en formato de lectura de redes neuronales.
- Determinar la temporada con más cantidad de precipitación y la temporada con menos precipitación.
- Determinar el número de capas y redes neuronales óptimo para la predicción de la cantidad de precipitación en el día.

CAPITULO II

ANTECEDENTES INVESTIGATIVOS

Después de haber realizado una investigación bibliográfica del tema investigado, se pueden destacar los siguientes trabajos de investigación:

En el trabajo denominado “Modelado y Predicción del Fenómeno El Niño en Piura, Perú mediante Redes Neuronales Artificiales usando Matlab” (Jiménez-Carrión et al., 2018) se menciona la aplicación de redes neuronales artificiales con los datos meteorológicos de precipitación, temperatura superficial del mar y velocidad del viento. La finalidad del proyecto fue modelar y predecir el fenómeno denominado El Niño con un periodo de anticipación de seis meses. La investigación se realizó en Piura, Perú mediante la realización de un análisis preliminar de los datos para establecer el nivel de correlación de las variables. Posteriormente se diseñó un modelo en base al uso de redes neuronales mediante el software MatLab que permite simular el comportamiento de las precipitaciones en la zona. La metodología utilizada consta de cinco pasos mismos que son: recolección de los datos, tratamiento preliminar, modelado de la red, creación del código y análisis de estabilidad de la red. Como resultado se tiene que las redes neuronales son una técnica confiable al momento de determinar un patrón de comportamiento de las precipitaciones. Se logró predecir el fenómeno logrando una tasa de efectividad del 98.4% en la fase de prueba y del 100% en la predicción que se realizó en el año 2016.

En el artículo denominado “Predicción de caudales medios diarios en la cuenca del Amazonas aplicando redes neuronales artificiales y el modelo neuro difuso ANFIS” (Chacón et al., 2016) se muestran los resultados de la implementación de una red neuronal y el modelo neuro difuso ANFIS. Este modelo fue utilizado para la predicción de caudales medios en la cuenca del Amazonas. Para esta investigación se aplicó la metodología KDD Knowledge Discovery in Databases con los datos obtenidos de la estación hidrológica del observatorio SO HYBAM. También se usó la información climatológica del Instituto de Investigación Internacional de Clima y Sociedad. Para la elaboración de la simulación se utilizó el software Matlab que permitió evaluar el comportamiento de ambos contextos. Con este estudio se demostró que al usar técnicas de inteligencia artificial se pueden conseguir coeficientes de correlación que sobrepasan el 97%. Según la investigación propuesta se puede concluir que estos

modelos son adecuados para la predicción, debido a su capacidad predictiva a comparación con las técnicas convencionales de tipo lineal.

En el artículo denominado “Kp forecasting with a recurrent neural network” (Sexton & Ma, 2019) se intenta predecir el índice Kp más allá de las predicciones de 1 y 4 horas clásicas. Se modula una red neuronal recurrente con tres décadas de datos históricos del observatorio virtual Omni de la Nasa y se pronostica el Kp con el margen de predicción de hasta 24 horas. Para esta investigación se utilizó el software Matlab. El modelo de percepción multicapa usado, tiene entradas compuestas por el Kp para un tiempo determinado, así como con diferentes conjuntos de seis parámetros del viento solar. El modelo consta de un conjunto de entradas, una capa oculta no lineal con 28 neuronas y una capa de salida lineal que predice Kp. La finalidad de esta investigación es probar que combinación de parámetros del viento y del campo magnético interplanetario da un mejor rendimiento. La investigación demuestra que la arquitectura de red neuronal propuesta es viable para pronosticar el índice Kp durante momentos de baja actividad solar. La red neuronal propuesta se benefició de la abundancia de datos históricos, sin embargo datos desequilibrados pueden conducir a predicciones equívocas.

En la investigación denominada “Predicting Weather Forecast Uncertainty with Machine Learning” (Scher & Messori, 2018) se menciona que las predicciones meteorológicas son inherentemente inciertas. Los pronósticos solo se consideran valiosos si se le asigna una estimación de la incertidumbre. En la actualidad el método más adecuado para la estimación de pronósticos es producir un conjunto de simulaciones meteorológicas numéricas. Las simulaciones mencionadas normalmente son procesos muy costosos. En esta investigación se evalúa si las técnicas de aprendizaje automático pueden proporcionar un enfoque alternativo para predecir la incertidumbre del pronóstico del tiempo. Se propone un método basado en aprendizaje profundo con redes neuronales convolucionales artificiales basada en pronósticos meteorológicos pasados. La principal limitación en el rendimiento de este método puede ser en el número de previsiones pasadas disponibles para entrenar el algoritmo de aprendizaje automático.

En el artículo denominado “Prediction of Rainfall in Australia Using Machine Learning” (Sarasa-Cabezuelo, 2022) menciona que el estudio de los fenómenos

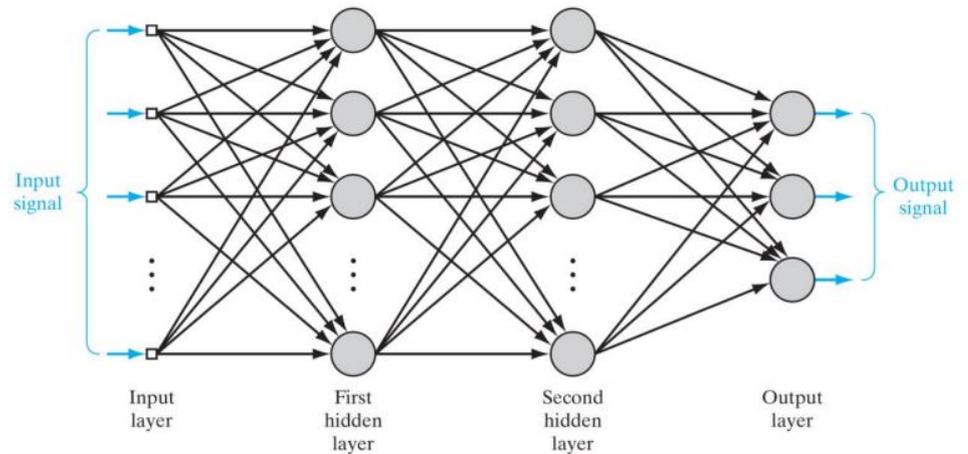
meteorológicos se considera un área con una gran cantidad de datos. Además, se menciona que es difícil realizar predicciones sobre eventos que ocurren debido al elevado número de variables. Para estos estudios generalmente se utilizan modelos probabilísticos que ofrecen predicciones con un margen de error. Este artículo describe un estudio exploratorio del uso del aprendizaje automático para hacer predicciones sobre el fenómeno de la lluvia. Para ello, se tomó como ejemplo un conjunto de datos que describen las mediciones recogidas sobre las precipitaciones en las principales ciudades de Australia. El periodo analizado pertenece a los últimos 10 años y se aplicaron algunos de los principales algoritmos de aprendizaje automático (knn, decision tree, random forest y redes neuronales). Los resultados muestran que el mejor modelo se basa en redes neuronales.

2.1. Redes neuronales

Se puede definir a las redes neuronales como un sistema que ayuda a establecer la relación entre entradas y salidas. Estas redes se basan en la estructura del sistema nervioso, se diferencian de la computación tradicional puesto que no usan un algoritmo secuencial. El comportamiento de estas redes es como un cerebro humano, procesan información en paralelo, teniendo la posibilidad de ilustrar y generalizar situaciones no incluidas en el proceso de entrenamiento (Acevedo et al., 2018).

Estas redes se utilizan principalmente para trabajos donde es difícil derivar restricciones lógicas explícitamente como en el caso del reconocimiento de patrones y análisis de predicción. También pueden ser utilizados en el análisis de relaciones no lineales de series de tiempo de parámetros meteorológicos como la precipitación, humedad, irradiancia entre otros. Las redes neuronales están compuestas por capas organizadas de unidades interconectadas también llamadas nodos. Estas están compuestas por una capa de entrada, una oculta y una de salida, sin embargo, cuando existen más de una capa oculta la red cambia y se convierte en red neuronal profunda (Rivas & Mazón, 2018).

Figura 1 Red neuronal multicapa



Fuente: Arana (2021)

2.2. Redes neuronales recurrentes

Las redes neuronales recurrentes (RNN) se consideran como un tipo de aprendizaje profundo que se basa en el trabajo de David Rumelhart (1986). Estas redes tienen la capacidad de procesar y obtener información procedente de datos secuenciales. Es decir, actividades como el análisis de video, procesamiento del lenguaje natural, entre otros, dependen de la capacidad de las redes recurrentes. En contraste con las redes artificiales que toman la independencia de los datos, las RNN capturan activamente sus dependencias secuenciales y temporales (Arana, 2021).

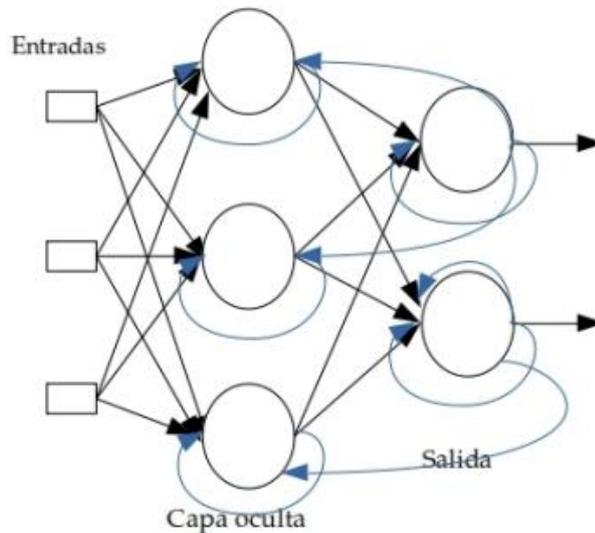
Según Habi y Messer (2020) las redes neuronales recurrentes son una clase de red neuronal que procesa datos secuenciales y destacan tanto en la clasificación de secuencias como en el mapeo. Las RNN, a diferencia de la red neuronal convencional, procesan entradas secuenciales al tener un estado interno que depende de todas las entradas anteriores. Uno de sus atributos más significativos es la capacidad de compartición de sus parámetros.

Este tipo de redes neuronales son más sofisticadas que las clásicas, sobre todo aplicables para modelar relaciones de datos de series temporales. La característica particular es que pueden procesar múltiples variables al realizar secuencias adecuadas para las capas de ingreso. Pueden extraer características de tiempo y sobre todo derivar en

predicciones temporales. Son resistentes a ruido, es decir datos que tengan un comportamiento atípico (Mohd et al., 2020).

Las redes neuronales recurrentes usualmente incrementan la estructura de su red multicapa convencional mediante la adición de ciclos que conectan nodos adyacentes o pasos de tiempo. Se utilizan para evaluar las propiedades de los datos actuales con respecto a los datos del tiempo pasado. Estas redes son capaces de efectuar el tratamiento de secuencias, la continuación de trayectorias de predicción no lineal y modelación de sistemas dinámicos (Arana, 2021).

Figura 2 Estructura de una red neuronal recurrente



Fuente: Llumitasig (2021)

2.3 Aplicación de redes neuronales recurrentes

Las redes neuronales recurrentes tienen algunas aplicaciones relacionadas con el procesamiento de secuencias mismas que se describen a continuación:

- **Predicción de series temporales:** es una aplicación muy utilizada en la actualidad donde conociendo los valores pasados de una o más variables, esta red debe predecir el posible valor futuro, se pueden usar series económicas o en fenómenos naturales, así también melodías que se encuentran sin terminar (Pontigo, 2020).

- Procesamiento del lenguaje humano, ecualización de canales digitales, codificación del habla, reconocimiento del habla, inferencia gramatical, control de sistemas (Guerrero, 2020).

2.4. Software Matlab

Su nombre es una abreviatura de “MATrix LABoratory”, es un software que permite realizar diversos cálculos numéricos con el uso de vectores y matrices. También se puede utilizar con números escalares ya sean reales o complejos, cadenas de caracteres y estructuras complejas de información. Matlab utiliza un lenguaje de alto rendimiento para el cálculo de valores técnicos, también se considera un entorno y un lenguaje de programación. Este software permite elaborar herramientas propias reutilizables, conocidos como M-archivos generados con código Matlab (Fernández, 2018).

Matlab ofrece un entorno de programación que contribuye al fácil manejo de matrices de datos, el procesamiento de señales y la construcción de algoritmos de aprendizaje automático. Permite gestionar una gran cantidad de datos que se pueden trabajar con machine learning, redes neuronales, Deep learning, visión artificial y conducción autónoma (Rueda, 2020).

Este software permite crear redes neuronales a través de modelos previamente generados o implementando modelos para servidores y dispositivos integrados. Matlab ayuda a integrar los resultados en sus aplicaciones existentes, contribuye a la automatización de los modelos de redes neuronales en sistemas empresariales, clusters, nubes y dispositivos integrados (Betolaza, 2021).

Matlab usa una arquitectura única que permite el acceso a todas las propiedades de la red neuronal, de esta manera se puede modificar las entradas, capas, conexiones, pesos, entre otros. Una vez configurada la red acorde a las necesidades de programador se utilizarán las funciones disponibles en Matlab tales como simulación, entrenamiento, inicialización, entre otros (Cheje, 2019).

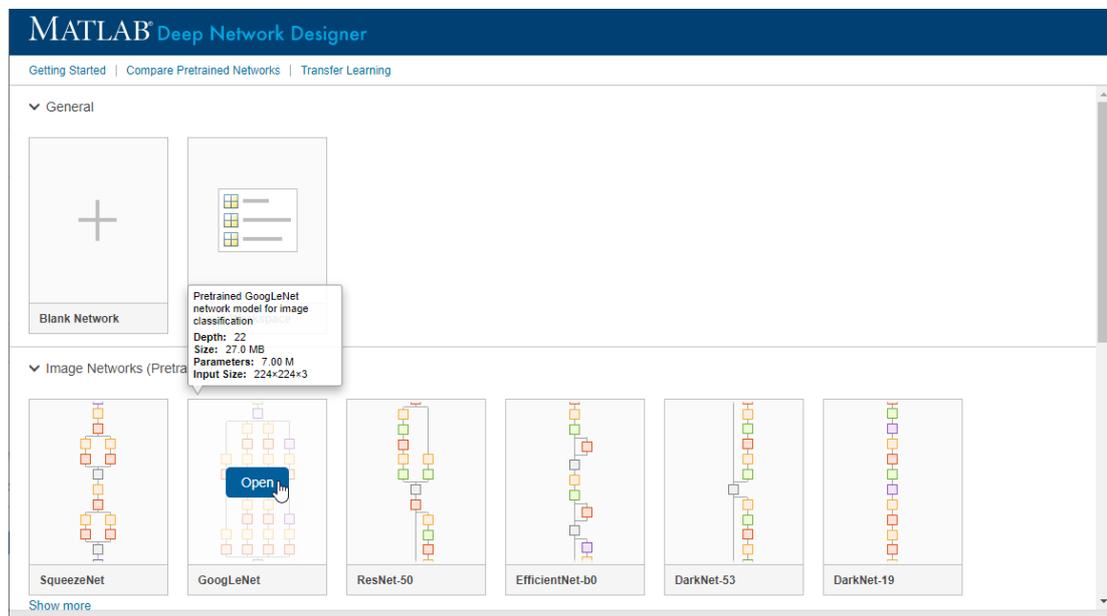
2.4.1. Deep network designer

Es una aplicación que permite diseñar, analizar y entrenar redes de forma gráfica, se pueden utilizar redes neuronales convolucionales y redes de memoria de corto y largo

plazo (LSTM). Puede realizar actividades de clasificación y regresión en series temporales, imágenes y datos de texto. También es capaz de crear estructuras de red como por ejemplo redes generativas antagónicas y redes siamesas por medio de la diferenciación automática, bucles de entrenamiento personalizado y peso compartido (MathWorks, 2022).

La figura 3 muestra un ejemplo de cómo utilizar Deep Network Designer para adaptar una red Google Net preentrenada para clasificar una colección de imágenes. Este proceso se denomina aprendizaje por transferencia y suele ser mucho más rápido y sencillo que entrenar una nueva red.

Figura 3 *Deep Network Designer*



Fuente: MathWorks (2022)

2.4.2. Entrenar redes con Deep Network Designer

En la fase de entrenamiento de una red neuronal se utiliza un conjunto de datos o patrones que ayudan a determinar los parámetros que van a definir el modelo de red. Se calculan de manera iterativa, en base a los valores de entrenamiento, con la finalidad de minimizar el error cometido entre la salida obtenida por la red neuronal y la salida deseada (Salas, 2018).

La aplicación Deep Network Designer permite construir y entrenar redes neuronales profundas. Admite el entrenamiento de la red mediante datos de imágenes u objetos de almacén de datos. También se puede exportar una red no entrenada para entrenarla en la línea de comandos, por ejemplo, utilizando bucles de entrenamiento personalizados.

Para entrenar una red se deben seguir los siguientes pasos:

- Crear una red
- Importar datos
- Seleccionar las opciones de entrenamiento
- Entrenar la red
- Exportar red

Se puede crear una red de forma interactiva utilizando el diseñador de redes profundas, o importando una red desde el área de trabajo. También es posible seleccionar una red preentrenada desde la página de inicio de Deep Network Designer para el aprendizaje por transferencia. Se puede entrenar una gran variedad de redes, por ejemplo, redes de clasificación o regresión de imágenes, secuencias, datos numéricos, segmentación semántica y regresión de imagen a imagen. Para entrenar una red es posible utilizar la función `trainNetwork` en cualquier dato que pueda expresarse como objeto de almacén de datos (Ederra, 2019).

Figura 4 Proceso de entrenamiento



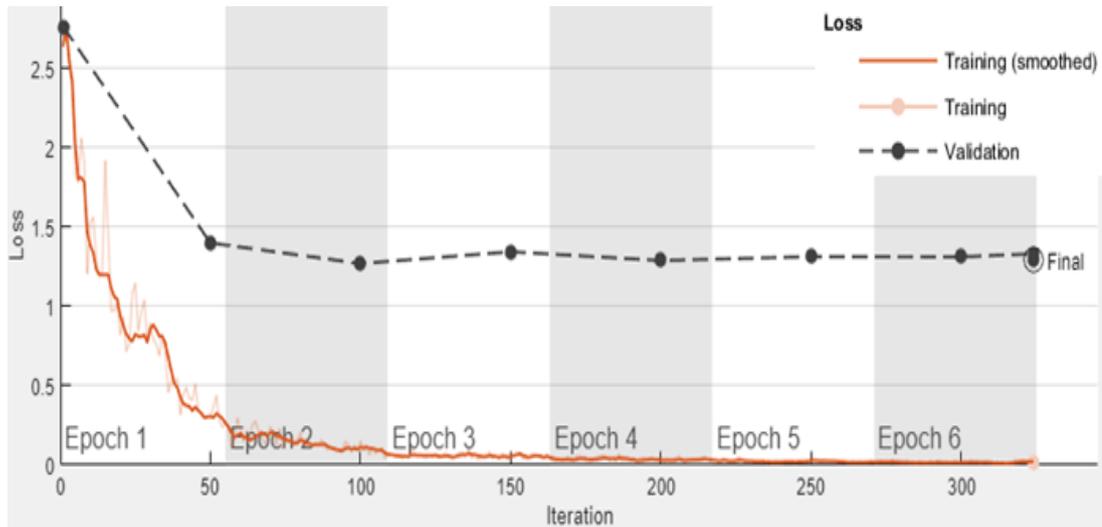
Fuente: MathWorks (2022)

Después del proceso de entrenamiento se tiene la fase de prueba o testeo, donde se comprueba si el modelo se ajusta a las particularidades de los patrones de entrenamiento. Para evitar problemas de entrenamiento se puede utilizar un segundo grupo de datos diferentes a los de entrenamiento, el grupo de validación, que permita controlar el proceso de aprendizaje. Habitualmente, los parámetros óptimos se obtienen mejorando (minimizando) alguna función de energía. Por ejemplo, minimizar el error cuadrático medio entre el valor de salida y el valor real esperado (Salas, 2018).

2.4.3. Datos de validación o Validation data

En Deep Network Designer, es posible importar datos de validación para utilizarlos durante el entrenamiento de una red. Los datos de validación tienen la información que utiliza la red para actualizar los pesos y los sesgos durante el entrenamiento. Como no se utilizan directamente estos datos, son útiles para evaluar la verdadera precisión de la red. Se pueden controlar las métricas de validación, como la pérdida y la precisión, para valorar si la red se está ajustando en exceso o en defecto (Beale et al., 2020).

Figura 5 Datos de validación



Fuente: MathWorks (2022)

En Deep Network Designer, se pueden importar datos desde un almacén de datos en el espacio de trabajo o dividiendo una parte de los datos de entrenamiento para utilizarlos como datos de validación. Los datos se dividen en conjuntos de validación y de entrenamiento una vez, antes del entrenamiento, este método se denomina validación por retención (Beale et al., 2020).

2.4.4. Datos de entrenamiento o Data Train

Los datos de entrenamiento son generalmente un conjunto de datos extremadamente grande que son utilizados para entrenar un modelo de aprendizaje automático. Estos datos se usan en los modelos de predicción que utilizan algoritmos de aprendizaje automático. Existen dos modelos de aprendizaje automático: supervisados donde los datos de entrenamiento están etiquetados y no supervisados donde los datos no están etiquetados y son utilizados para entrenar modelos (Janalta Interactive , 2022).

Los datos de entrenamiento son un conjunto inicial de datos que se utilizan para ayudar a un programa a comprender un fenómeno estadístico. Ayudan a las redes neuronales para aprender y producir resultados sofisticados. Además, pueden complementarse con conjuntos de datos posteriores llamados conjuntos de validación y prueba (Janalta Interactive , 2022).

Es el material a través del cual la computadora aprende a procesar la información para generar un aprendizaje automático. Utiliza algoritmos que tratan de imitar las capacidades del cerebro humano para recibir diversas entradas y sopesarlas, a fin de producir activaciones en el cerebro, en las neuronas individuales (Janalta Interactive , 2022).

Los datos de entrenamiento se pueden estructurar de diferentes maneras como árboles de decisión secuenciales. La idea es que debido a que el programa de aprendizaje automático es tan complejo y sofisticado, utiliza un entrenamiento iterativo (Janalta Interactive , 2022).

2.4.5. Data de testeo o Data test

Los datos de prueba o testeo son los datos que guardamos para demostrar si el modelo que generamos a partir de los datos de entrenamiento funciona. Se usan para verificar si las respuestas predichas por el modelo para un caso totalmente nuevo son apropiadas o no (Recuero de los Santos P., 2022).

Es importante que el conjunto de datos de prueba tenga la suficiente magnitud como para generar resultados estadísticamente significativos y que sean representativos del conjunto de datos global. Normalmente los datos se suelen repartir en un 70% de datos de entrenamiento y un 30% de datos de testeo, pero se puede variar la proporción según el caso. Es sustancial estar siempre conscientes de que hay que impedir el “overfitting” o el sobreajuste que sobreentrena al modelo (Recuero de los Santos, 2022).

2.4.6. Función de activación en el campo de redes neuronales

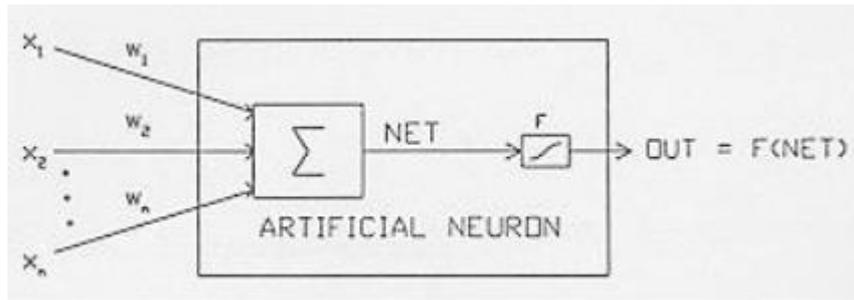
La función de activación fue diseñada para intentar simular las características del funcionamiento de una neurona biológica en una neurona artificial. Para la simulación mencionada se aplica un conjunto de entradas a la neurona artificial, las cuales representarán una salida de otra neurona. Para cada entrada se multiplica por su ponderación o peso correspondiente semejante al grado de conexión de la sinapsis. Todas las entradas ponderadas se suman para determinar el nivel de excitación o activación de la neurona (Olabe, 2008).

Una representación vectorial del funcionamiento básico de una neurona artificial se indica según la siguiente expresión de la ecuación 1.

$$NET = [X] * [W] \quad (1)$$

Siendo NET la salida, X el vector de entrada y W el vector de ponderaciones.

Figura 6 Modelo neurona artificial



Fuente: Xavier Basogain Olabe (2008)

Como se observa en la figura 6, normalmente la señal de salida NET se resuelve por una función de activación F para producir la señal de salida de la neurona OUT. La función (F) puede ser una función lineal, una función umbral o una función no lineal. La función no lineal simula con mayor exactitud las características de transferencia no lineales de las neuronas biológicas. Entre las funciones (F) más utilizadas son la función Sigmoid y Tangente hiperbólica (Olabe, 2008).

2.4.7. Función de activación Tangente hiperbólica (Tanh)

La función de activación se encarga de devolver una salida a partir de un valor de entrada. La función tangente hiperbólica transforma los valores introducidos a una escala (-1,1). Los valores altos tienden de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a -1 (Calvo, 2018).

Tabla 1 Ecuación de la función de activación Tangente hiperbólica (Tanh)

$f(x) = \frac{2}{1 + e^{-2x}} - 1$ <p>(2)</p>	<p>Características:</p> <ul style="list-style-type: none"> • Saturación y mata el gradiente. • Lenta convergencia. • Centrada en 0. • Esta acotada entre -1 y 1.
---	--

-
- Se utiliza para decidir entre una opción y la contraria.
 - Buen desempeño en redes recurrentes.
-

Fuente: Calvo (2018).

2.4.8. Función de activación Rectified Linear Unit (Relu)

La función Relu transforma los valores introducidos, anulando los valores negativos y dejando los positivos tal y como entran (Calvo, 2018).

Tabla 2 Ecuación de la Función de activación Rectified Linear Unit (Relu)

$f(x) = \max(0, x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$ <p>(3)</p>	<p>Características:</p> <ul style="list-style-type: none"> • Activación Sparse – solo se activa si son positivos. • No está acotada. • Se pueden morir demasiadas neuronas. • Se comporta bien con imágenes. • Buen desempeño en redes convolucionales
--	---

Fuente: Calvo (2018).

2.4.9. LSTM (Long Short-Term Memory)

Las LSTM son un tipo especial de redes recurrentes, en la cual la característica principal es que la información puede permanecer introduciendo bucles en el diagrama de la red. La capa LSTM recuerda estados previos y utiliza esta información para decidir cuál será el siguiente estado. Esta característica las hace muy útiles para manejar series cronológicas. Las LSTM pueden aprender dependencias largas, por lo que se podría decir que tienen una memoria a más largo plazo (Fierro, 2020).

2.4.10. Regresión lineal en el campo de redes neuronales

Para los problemas de predicción, los modelos de redes neuronales son aplicados en forma similar a los modelos estadísticos, pero con la limitación de no poder realizar un análisis explicativo del modelo ajustado. Entre las propiedades que se destacan en el uso de las redes neuronales con respecto a los modelos estadísticos clásicos, esta su aplicación sin la necesidad de considerar el cumplimiento de supuestos teóricos. Las redes neuronales en la estadística son definidas como técnicas no paramétricas. La regresión lineal es utilizada al finalizar la arquitectura de la red neuronal para pronosticar valores (Menacho, 2014).

2.4.11. Normalización de datos

El parámetro de normalización (o factor de escala) mide la anchura de la gaussiana, que equivaldría al radio de influencia de la neurona en el espacio de las entradas. A mayor región que la neurona domina en torno al centroide, la normalización es más amplia. Una vez ajustados los pesos es necesario re normalizar el vector de análisis. El ajuste de pesos y normalización de las entradas es el efecto de mover el vector inicial hacia el vector de entrada (Olabe, 2008).

La normalización de la entrada se va a realizar con la inclusión de una nueva capa de neuronas que conste de tantas neuronas como tenga la capa de entrada. La red tiene la capacidad de generalizar a partir de los casos mostrados y proveer una respuesta a una entrada que nunca le ha sido formulada (Olabe, 2008).

2.4.12. Error RMSE

El Error de raíz cuadrada media (RMSE) es la desviación estándar de los valores residuales o los errores de predicción. Estos valores residuales son una medida del camino de los puntos de datos de la línea de regresión. RMSE es una medida que indica el nivel de dispersión de estos valores residuales. En otras palabras, revela el nivel de reunión de los datos en la línea de mejor ajuste (Velasquez, 2011).

CAPITULO III MARCO METODOLÓGICO

3.1. Ubicación

El estudio se realizó gracias a los datos proporcionados por la estación meteorológica de Querochaca, ubicada en los predios de la Universidad Técnica de Ambato. Específicamente se encuentra en el cantón Quero de la provincia de Tungurahua, con coordenadas 1394055.59, 3045272.81, su código es M258.

Figura 7 Estación Meteorológica de Querochaca



Fuente: Red Hidrometeorológica de Tungurahua (2022)

Esta estación suministra los datos de los parámetros como: Precipitación (mm), Temperatura (°C), Humedad relativa (%), Velocidad del viento (m/s), Dirección del viento (°), Presión atmosférica (hPa). De estos valores, se utilizaron los datos de precipitación para generar los pronósticos propuestos, a través de redes neuronales artificiales.

3.2. Equipos y materiales

- Computadora
- Datos de precipitación de 7 años
- Matlab

3.3. Tipo de investigación

La investigación se realizó a través de una metodología cuasi experimental. Se valió del uso de redes neuronales artificiales para generar pronósticos de precipitaciones mediante el auto aprendizaje y entrenamiento en base a datos obtenidos. Estos datos se obtuvieron por investigación libre en las redes públicas del Instituto Nacional de Meteorología e Hidrología del Ecuador.

3.4. Recolección de información

A través de redes públicas del Instituto Nacional de Meteorología e Hidrología del Ecuador se recolectaron datos de precipitación de 7 años, desde el 2015 hasta el 2021 de la estación mencionada previamente en el documento.

3.5. Hipótesis

H0: Mediante los datos de precipitación de la estación meteorológica no se puede predecir precipitaciones utilizando redes neuronales artificiales.

H1: Mediante los datos de precipitación de la estación meteorológica si se puede predecir precipitaciones utilizando redes neuronales artificiales.

3.6. Estructura de la Información DATASET

Los datos de precipitación obtenidos fueron estructurados en 3 conjuntos específicos como se observa en la Tabla 3. Primeramente, los datos de entrenamiento, que contienen la mayor parte de valores, es decir el 60% que son 1533 valores, los que permiten el adiestramiento de la red neuronal. El 20% de los datos se destinó para el proceso de validación. Estos datos sirven para contrastar los valores generados por una red neuronal entrenada, y tener un parámetro cuantitativo y cualitativo para aplicar mejoras al modelo general. Finalmente, el 20% restante, se lo destinó para el testeo. Estos datos finales sirven para la validación final del modelo, cuando este ya tiene las modificaciones respectivas obtenidas de la experimentación en el proyecto.

Tabla 3 División del DATA SET

Tipo de datos	Porcentaje del total data set	Cantidad de datos por año	Cantidad de datos totales
Entrenamiento	60%	219	1533
Validación	20%	73	511
Testeo	20%	73	511

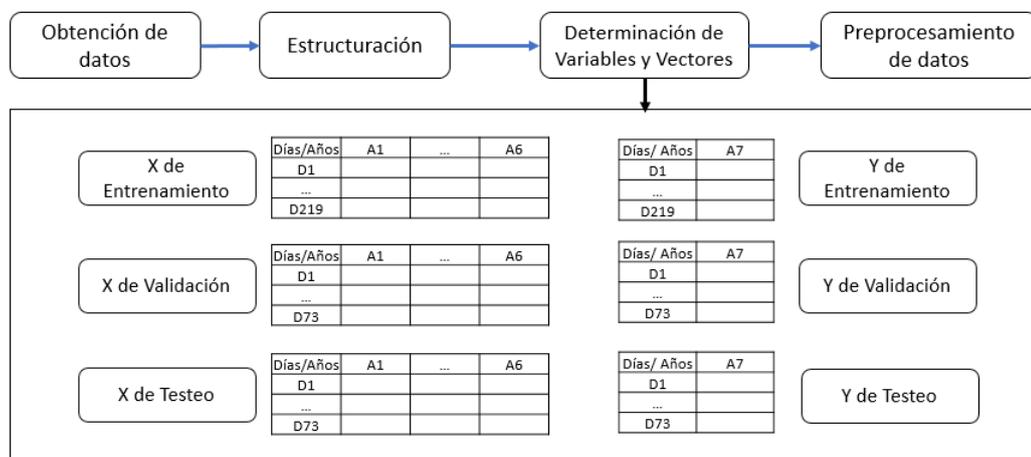
Fuente: Elaboración propia

Cada uno de estos grupos del dataset, fueron almacenados en archivos individuales organizados en 7 columnas, las cuales representan los años. Además, dependiendo del tipo de dato, también se estructuraron filas, las que vienen representando los días del año analizados.

3.7. Variables

Los datos de los 7 años se organizaron en dos variables X y variable Y, donde la variable X contiene los datos de los años 2015 hasta el 2020, mientras que la variable Y contienen los datos del año 2021. Los datos de la variable X son los que sirven como variable independiente, y constituyen un vector de 6 columnas. La variable Y es la dependiente del vector X. La variable Y es un vector de 1 columna. Con las variables definidas se procedió a establecer los vectores detallados en la Figura 8.

Figura 8 Datos y subprocesos de organización



Fuente: Elaboración propia

3.8. Preprocesamiento de datos

El preprocesamiento de datos se lo realizó a través de la normalización de datos. Este proceso consiste en obtener el vector de medias aritméticas (μ) de los datos de entrenamiento. Posteriormente, se calcula el vector de desviaciones estándar (σ). A continuación se procede a realizar el cálculo del vector final de datos estandarizados, en la cual se toman en cuenta los valores obtenidos previamente, todo lo mencionado se observa en la *Ecuación 4*.

$$X_{ent.est.} = \frac{(X_{ent.} - \mu)}{\sigma} \quad (4)$$

Donde $X_{ent.}$ es el vector de los datos de entrenamiento de X, $X_{ent.Est.}$ es el vector de datos estandarizados de X. Además μ es el vector de medias y σ el vector de desviaciones estándar.

De la misma manera se procedió con los datos de la variable Y de entrenamiento. En este caso se generaron datos escalares de la media y de la desviación estándar, debido a que son valores de un solo año, el cual es el séptimo.

El preprocesamiento se aplicó también a los datos de validación y testeo, tomando como base los datos de media y desviación estándar previamente calculados con los vectores de entrenamiento. Todo esto con el objetivo de tener un parámetro fijo de correspondencia a la tendencia y análisis.

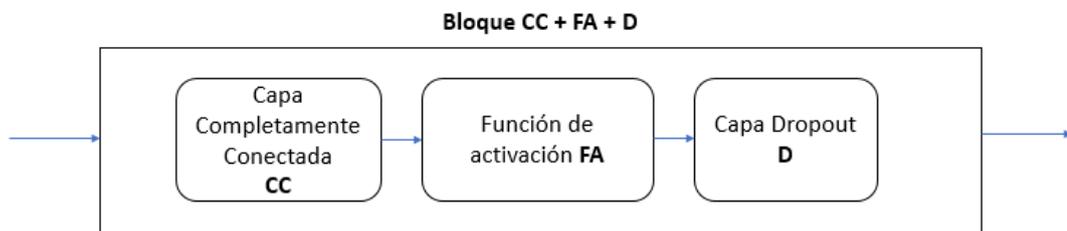
3.9. Arquitectura base

El diseño de la red neuronal depende esencialmente de la arquitectura utilizada. En el presente trabajo se utilizaron varias capas que tienen su importancia en base al papel que cumplen en el pronóstico. En perspectiva general se colocaron 6 formas de capa, las cuales son:

- **Ingreso (Input):** Esta capa se utiliza para ingresar el vector con la variable independiente. En el presente trabajo se ingresaron los datos del vector X de entrenamiento con un valor de 6, por los seis años de análisis. Debido a que se trabaja en la red neuronal recurrente, es necesario que este ingreso sea a través de una capa input de secuencia.

- **LSTM:** Es la capa específica que genera la recurrencia de la red neuronal, debido a que toma en consideración los valores previos para desarrollar la estimación o predicción.
- **Capa Dropout:** Se colocó esta capa para evitar tener valores que se alejen del contexto de la predicción realizada.
- **Bloque Completamente Conectada + Función de activación + Dropout (CC + FA + D):** En esta sección es donde se dio la variabilidad experimental aplicada en el proyecto, debido a que se trabajó con uno, con dos y con tres bloques. Cada bloque cuenta con su respectiva Capa Completamente Conectada. En la cual se colocó la cantidad de neuronas a utilizar, donde dependiendo de la experimentación se utilizaron 10, 50, 100, 500 y 1000 neuronas. El bloque tiene una función de activación, la cual puede ser Tanh o Relu; y finalmente tiene una capa dropout. La secuencia de las capas colocadas en el bloque actual se observa en la Figura 9.

Figura 9 Capas presentes en el Bloque CC + FA + D

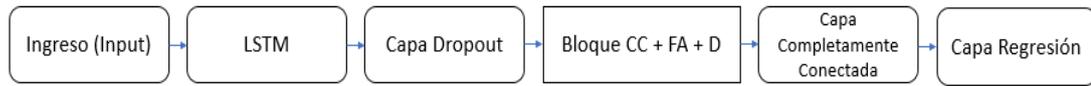


Fuente: Elaboración propia

- **Capa Completamente Conectada Salida:** En esta capa se ingresó el valor específico de uno, debido a que al aplicar la regresión lineal se obtuvo un valor de pronóstico por cada conjunto de entrada. Es prácticamente la capa output de la arquitectura es decir la salida.
- **Capa de Regresión:** Esta capa se la usó para aplicar el criterio de regresión lineal del pronóstico de precipitación.

La secuenciación de las capas se observa esquematizada en la Figura 10.

Figura 10 Arquitectura general de la red neuronal



Fuente: Elaboración propia

3.10. Método de entrenamiento de la red neuronal

El método de entrenamiento tiene una alta relevancia con respecto a los resultados que se van a generar. En perspectiva general, de inicio se utilizaron los parámetros recomendados por el software utilizado. Se destacan principalmente la presencia del optimizador ADAM, y el valor 50 en máximo de épocas de entrenamiento. Este último valor fue colocado después de realizar pruebas previas, y analizar la viabilidad del mismo. Existen otros parámetros adicionales implementados en el entrenamiento presentados en la Tabla 6.

Tabla 4 Parámetros definidos para método de entrenamiento

Parámetro	Valor o Tipo
Optimizador	Adam
Máximo de Épocas	50
Umbral de gradiente	1
Tasa de aprendizaje inicial	0,005
Periodo de caída de tasa de aprendizaje	125
Factor de caída de tasa de aprendizaje	0,2
Verbosa	0

Fuente: Elaboración propia

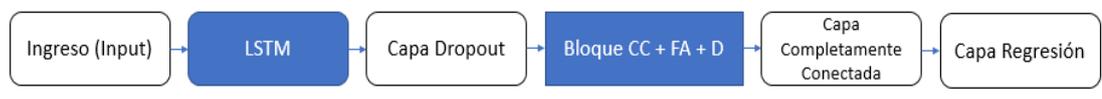
3.11. Experimentación y validación de la arquitectura

- **Experimentación y variación en las capas**

En la búsqueda del mejor modelo de red neuronal recurrente para el proyecto, se trabajó con variaciones en dos partes específicas de la arquitectura base como se observa en la Figura 11. Estas partes son la celda LSTM y el Bloque CC + FA + D. Se realizaron 348 variaciones iniciales, la cuales se ejecutaron tomando en cuenta

solamente el data set de entrenamiento y el data set de validación. El data set de testeo se utilizó solamente para la experimentación final.

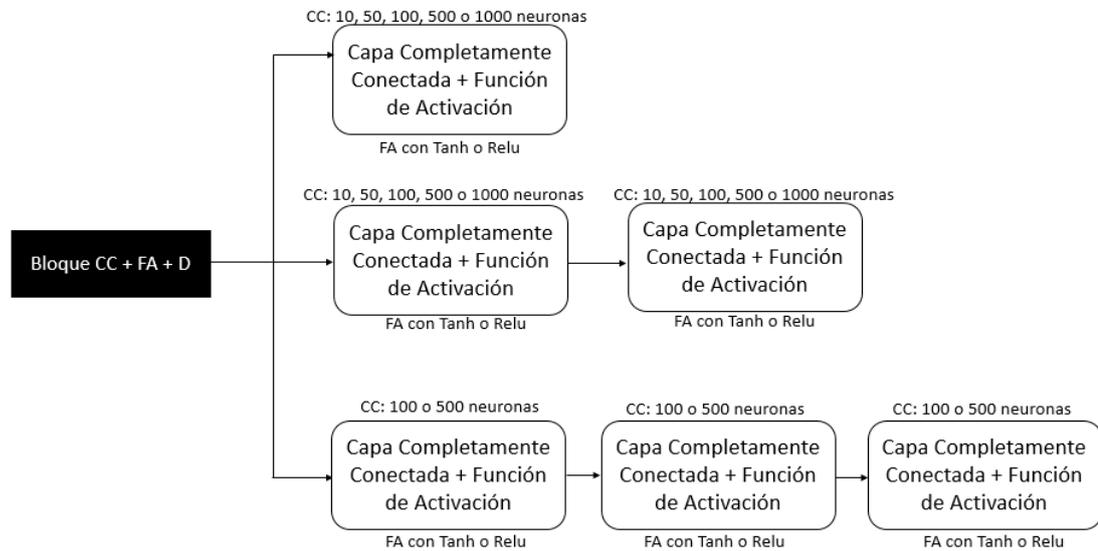
Figura 11 Capas de experimentación



Fuente: Elaboración propia

En la capa LSTM se trabajó con 10 y 100 unidades ocultas. En el bloque CC + FA + D se realizó la modificación en el tipo de función de activación, y en la cantidad de neuronas de la capa Completamente Conectada, tal como se detalla en figura 12.

Figura 12 Variabilidad en el Bloque CC + FA + D



Fuente: Elaboración propia

- **Estimación y pronóstico de datos**

Con la red neuronal entrenada, se procedieron a calcular los datos de la variable Y, a partir de los valores del vector X de validación estandarizado. Para realizar los cálculos se utilizó la función predecir, con los parámetros específicos del entrenamiento, y también el vector de ingreso ya mencionado.

3.12. Desestandarización de datos

Los datos obtenidos de la predicción con la red neuronal entrenada tienen un valor estándar ajustado, es decir bajo una media y desviación estándar específicos. Por esta

razón fue necesario aplicar un proceso de desestandarización. En este proceso se tomaron en consideración los valores de desviación estándar del vector Y de entrenamiento, también se utilizó el valor de media del mismo vector como se observa la ecuación 5.

$$Y_{des} = Y_{pron} * \sigma_{Y_{ent}} + \mu_{Y_{ent}} \quad (5)$$

Donde Y_{des} representa el valor del dato de Y desestandarizado, Y_{pron} representa el valor de la predicción realizada por la red entrenada, $\sigma_{Y_{ent}}$ representa la desviación estándar del vector Y de entrenamiento. Finalmente, $\mu_{Y_{ent}}$ es la media de Y de entrenamiento.

3.13. Validación del modelo

3.13.1. Graficación y cálculo de error

Con los datos obtenidos, se realizaron dos tipos de gráficas que permitieron tener un panorama general del comportamiento del modelo, estos gráficos son:

- **Comparación de valores (Pronóstico):** En esta gráfica se colocaron los datos del vector Y de validación, que son los datos reales; y los datos de generados por el modelo. Estos últimos son los datos obtenidos de la experimentación con la red neuronal con el proceso de desestandarización.
- **Histograma VR vs VO:** Se realizó un histograma de frecuencias donde se observa la diferencia del valor obtenido y el valor real.

Además, para tener una métrica de comparación del comportamiento del modelo, se procedió al cálculo de error RMSE. Esta toma en consideración el valor obtenido y el valor de validación, lo que se detalla en la ecuación 6.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_{des} - y_{validación})^2}{n}} \quad (6)$$

Donde y_{des} es el valor desestandarizado obtenido del modelo, $y_{validación}$ es el valor real que está ubicado en el vector Y validación, y n el número total de datos analizados.

3.13.2. Variaciones y mejoras a los modelos

De los 348 resultados obtenidos de la experimentación, se escogieron los mejores resultados de cada una de las combinaciones realizadas. Este proceso se lo realizó en

función del más bajo RMSE y de la adaptación de la gráfica con los valores pronosticados con respecto a los valores reales. De esta fase se obtuvo una lista preliminar de 25 modelos. Posteriormente se depuró la lista escogiendo los 5 mejores resultados, para poder aplicar variaciones específicas para mejorar aún más el modelo.

El criterio que tuvo más peso para escoger los mejores modelos fue la gráfica de pronóstico. Donde el observar una tendencia similar a la gráfica real permite mirar picos de tendencia, en contraste a gráficas con valores similares a la media aritmética de los datos.

Se aplicaron variaciones específicas a los modelos, tomando en cuenta cambios en los parámetros del método de entrenamiento. Específicamente se modificó el factor de dropout de las distintas capas, además de cambios en la cantidad de unidades ocultas de la capa LSTM. Incluso se aplicaron variaciones en la velocidad inicial de aprendizaje de la red y el número de épocas para las iteraciones. Estas modificaciones se aplicaron a los modelos sin tener una mejora sustancial en el RMSE, pero sí en la tendencia de gráfica. En esta última se generaron picos y valores cercanos al cero, lo cual se acerca más al comportamiento de la variable analizada.

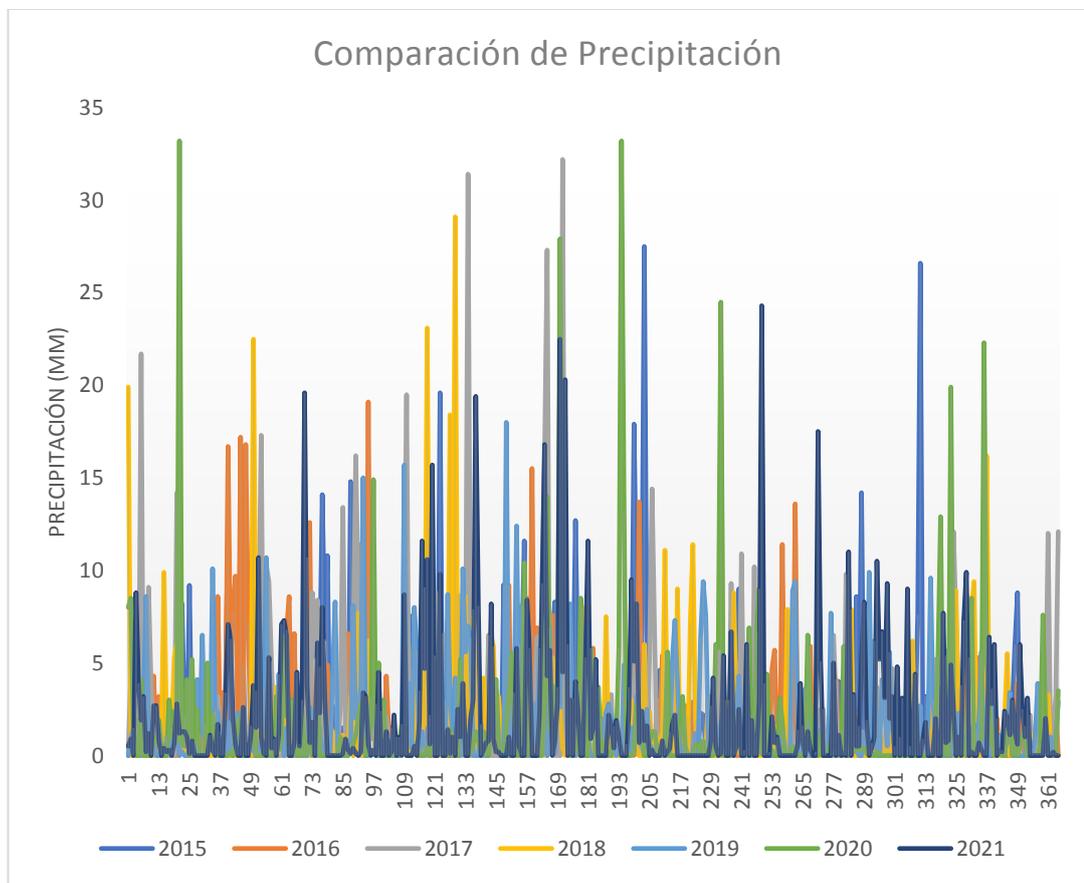
3.13.3. TESTEO

Con el modelo mejorado se realizó la experimentación y validación final, es decir se aplicó la red neuronal con la arquitectura final al dataset de testeo. En esta validación definitiva del modelo, se cargó el data set de testeo en forma de variable X de testeo, y como variable Y de testeo. La finalidad fue realizar la comparación final del resultado predicho por el modelo con respecto a los valores reales.

CAPITULO IV RESULTADOS

Para iniciar este capítulo, se presenta a continuación un gráfico comparativo de los datos de precipitación de los 7 años analizados en el presente proyecto. En la figura 13 se observa que los meses con mayor cantidad de precipitación son de junio a agosto, dado que los mayores valores de lluvia se presentan en el intervalo de los días 150 a 200. Los meses con menos cantidad de precipitación son diciembre y enero.

Figura 13 Comparación de Precipitación



Fuente: Elaboración propia

A continuación, en esta sección se presentan los resultados de los 5 modelos que se definieron en el capítulo anterior. Así también se presentan los resultados de las mejores aplicadas a cada arquitectura.

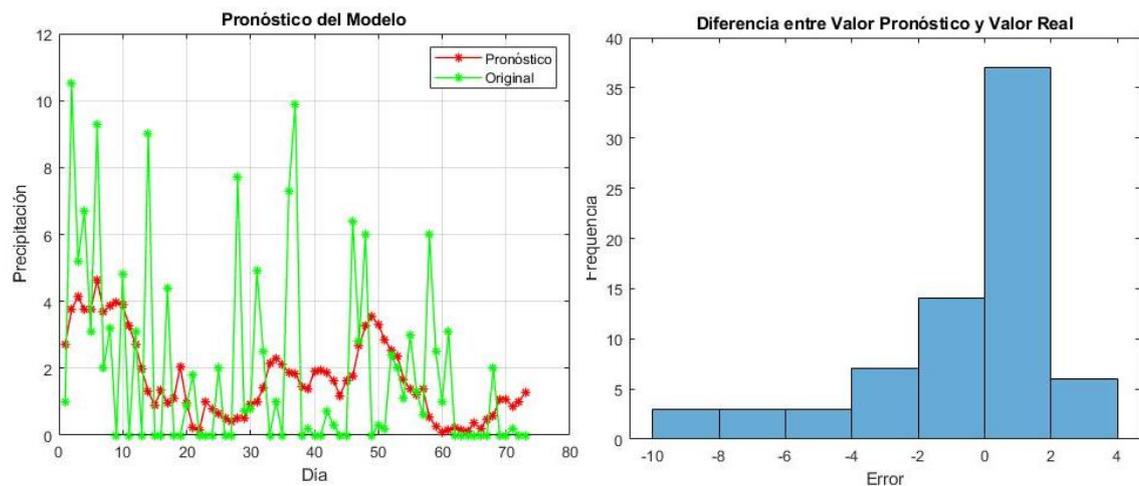
4.1. Modelo 1

Tabla 5 Arquitectura del Modelo 1

Capa / Parámetro	Cantidad de Neuronas / Valor
Unidades Ocultas – Capa LSTM	100
Capa completamente conectada – F. Activación Tanh	100
Capa completamente conectada – F. Activación Relu	500
Error RMSE	2,63

Fuente: Elaboración propia

Figura 14 Pronóstico generado por el modelo 1 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

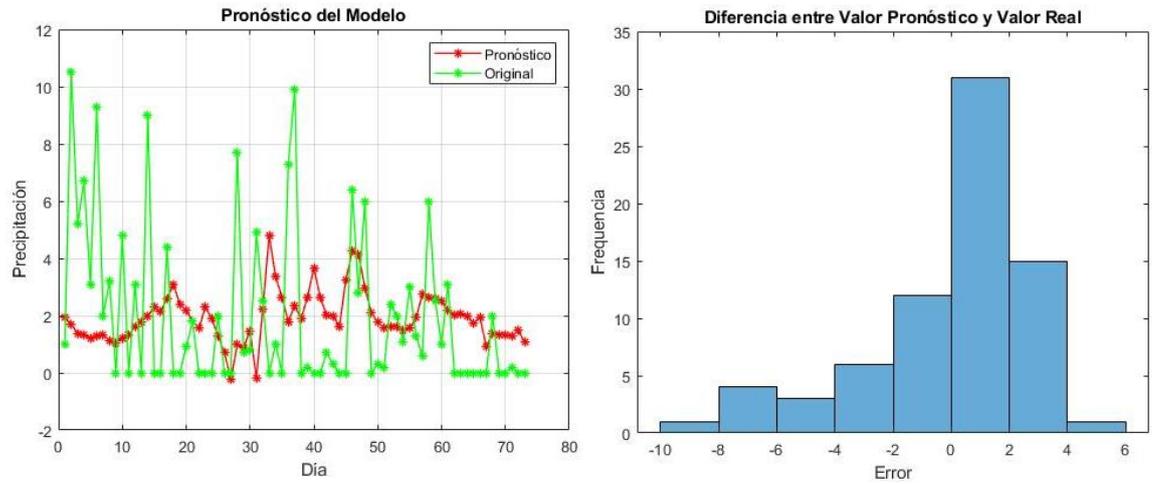
4.1.1 Variación 1

Tabla 6 Variación 1 aplicada a la arquitectura del Modelo 1

Capa / Parámetro	Original	Variación
Capa Dropout 1	0,2	0,1
Capa Dropout 2	0,2	0,4
Capa Dropout 3	0,2	0,4
Error RMSE	2,63	2,9219

Fuente: Elaboración propia

Figura 15 Pronóstico generado por la variación 1 del modelo 1 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

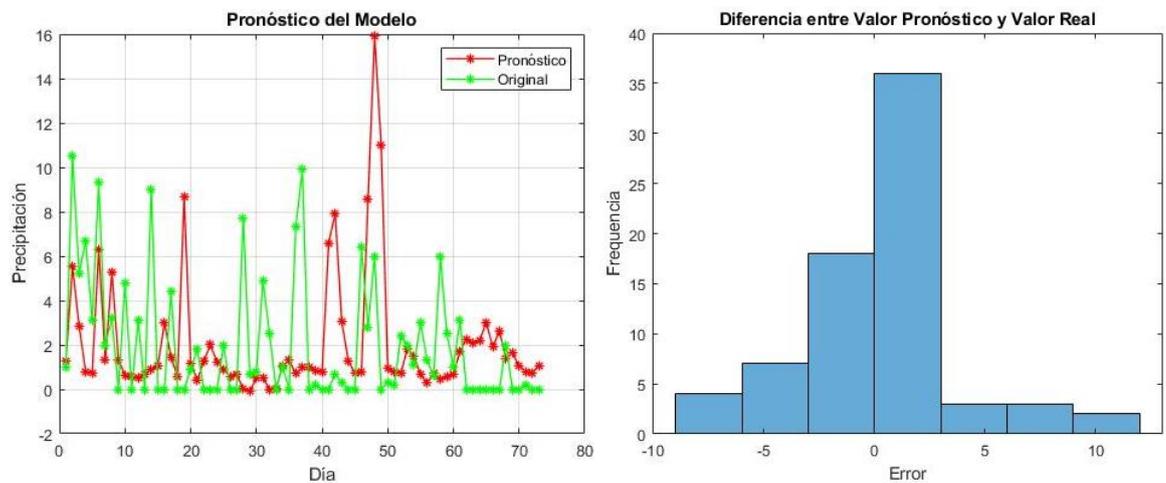
4.1.2 Variación 2

Tabla 7 Variación 2 aplicada a la arquitectura del Modelo 1

Capa / Parámetro	Original	Variación
Capa Dropout 1	0,2	0,1
Capa Dropout 2	0,2	0,4
Capa Dropout 3	0,2	0,4
Máximo de Épocas	50	500
Error RMSE	2,63	3,645

Fuente: Elaboración propia

Figura 16 Pronóstico generado por la variación 2 del modelo 1 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

El modelo 1 muestra como base de arquitectura a la capa LSTM con 100 unidades ocultas. Además, una capa completamente conectada con 100 neuronas que preceden a la función de activación Tanh, y una capa completamente conectada con 500 neuronas que preceden a la función de activación Relu. El error RMSE es relativamente bajo con un valor de 2,63; sin embargo, no se observan picos pronunciados en la gráfica. Los valores de precipitación que se obtienen como máximos bordean el valor de 4. Las variaciones se aplicaron principalmente en los valores de las capas dropout. Se aumentó el valor en las capas dropout posteriores a las funciones de activación a 0,4, y en la capa dropout posterior a LSTM, se disminuyó a 0,1. El comportamiento de la gráfica mejoró sustancialmente. Adicionalmente, en las épocas de análisis para las iteraciones, el valor se modificó a 500.

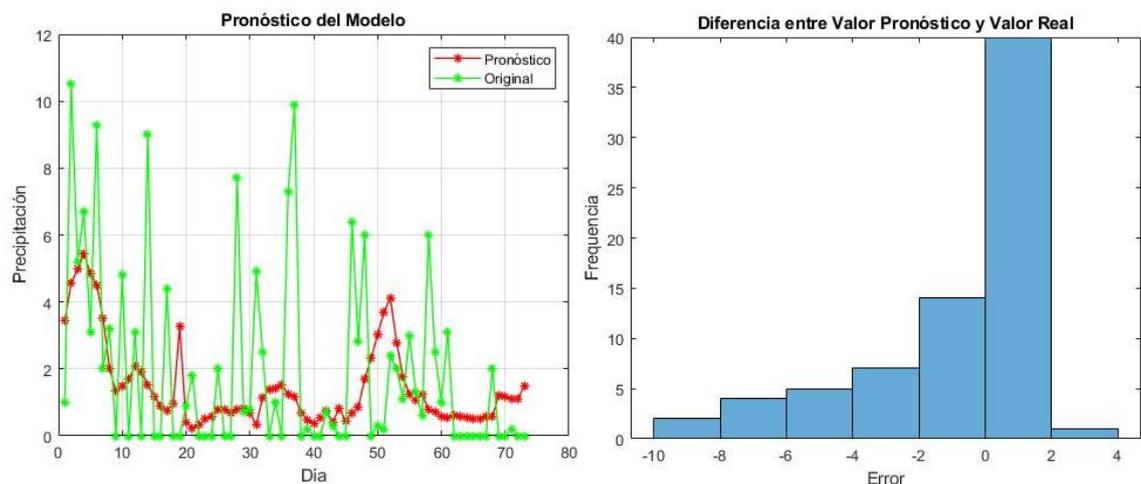
4.2. MODELO 2

Tabla 8 *Arquitectura del Modelo 2*

Capa / Parámetro	Cantidad de Neuronas / Valor
Unidades Ocultas – Capa LSTM	100
Capa completamente conectada – F. Activación Relu	100
Capa completamente conectada – F. Activación Relu	1000
Error RMSE	2,6336

Fuente: Elaboración propia

Figura 17 *Pronóstico generado por el modelo 2 en comparación al original, y Diferencia entre valor pronóstico y valor real*



Fuente: Elaboración propia

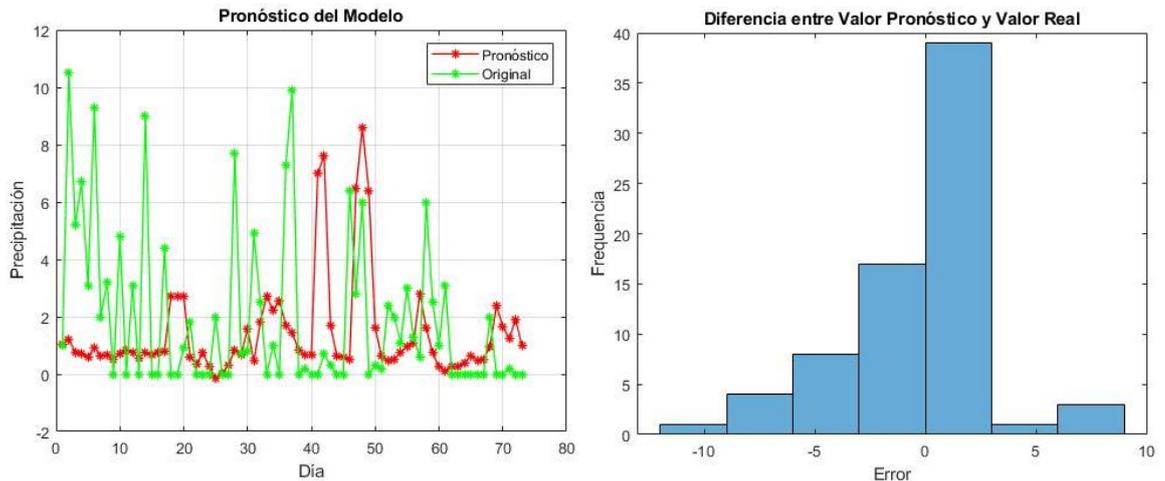
4.2.1. Variación 1

Tabla 9 Variación 1 aplicada a la arquitectura del Modelo 2

Capa / Parámetro	Original	Variación
Factor de caída de tasa de aprendizaje	0,2	1
Máximo de Épocas	50	500
Error RMSE	2,6336	3,3192

Fuente: Elaboración propia

Figura 18 Pronóstico generado por la variación 1 del modelo 2 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

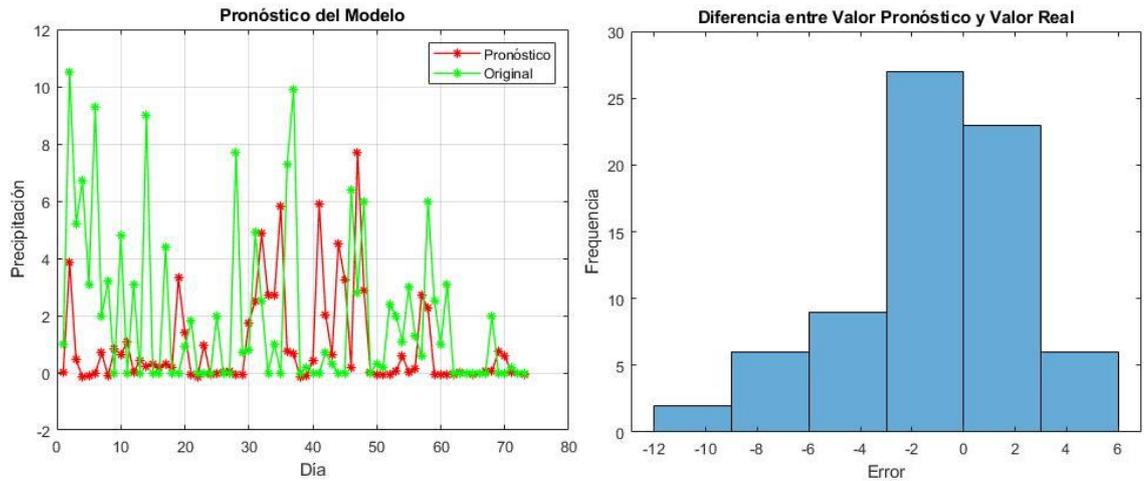
4.2.2 Variación 2

Tabla 10 Variación 2 aplicada a la arquitectura del Modelo 2

Capa / Parámetro	Original	Variación
Factor de caída de tasa de aprendizaje	0,2	1
Tasa de aprendizaje inicial	0,005	0,02
Máximo de Épocas	50	500
Error RMSE	2,6336	3,3387

Fuente: Elaboración propia

Figura 19 Pronóstico generado por la variación 2 del modelo 2 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

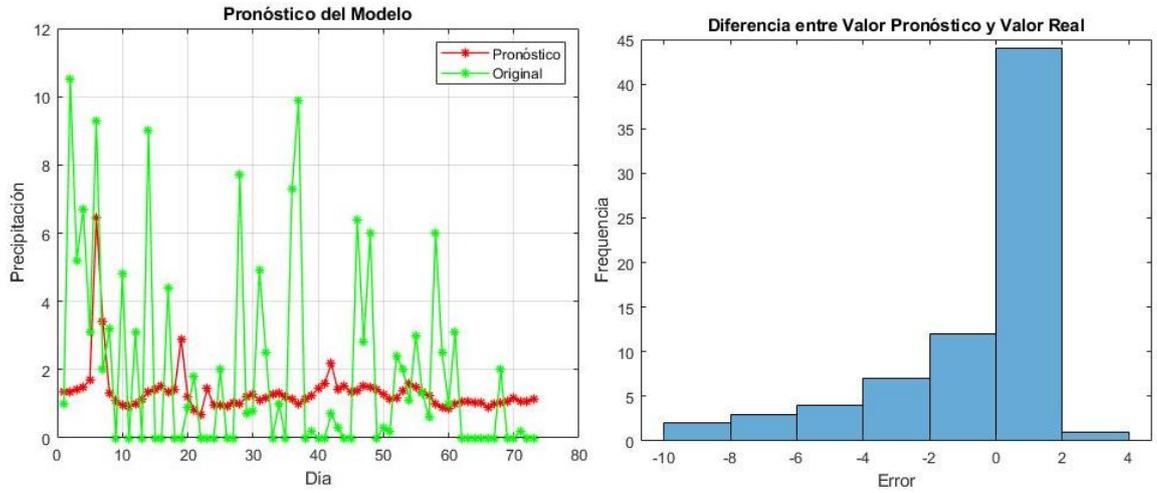
4.2.3. Variación 3

Tabla 11 Variación 3 aplicada a la arquitectura del Modelo 2

Capa / Parámetro	Original	Variación
Factor de caída de tasa de aprendizaje	0,2	1
Tasa de aprendizaje inicial	0,005	0,02
Máximo de Épocas	50	100
Capa Dropout 2	0,2	0,5
Capa Dropout 3	0,2	0,5
Error RMSE	2,6336	2,7464

Fuente: Elaboración propia

Figura 20 Pronóstico generado por la variación 3 del modelo 2 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

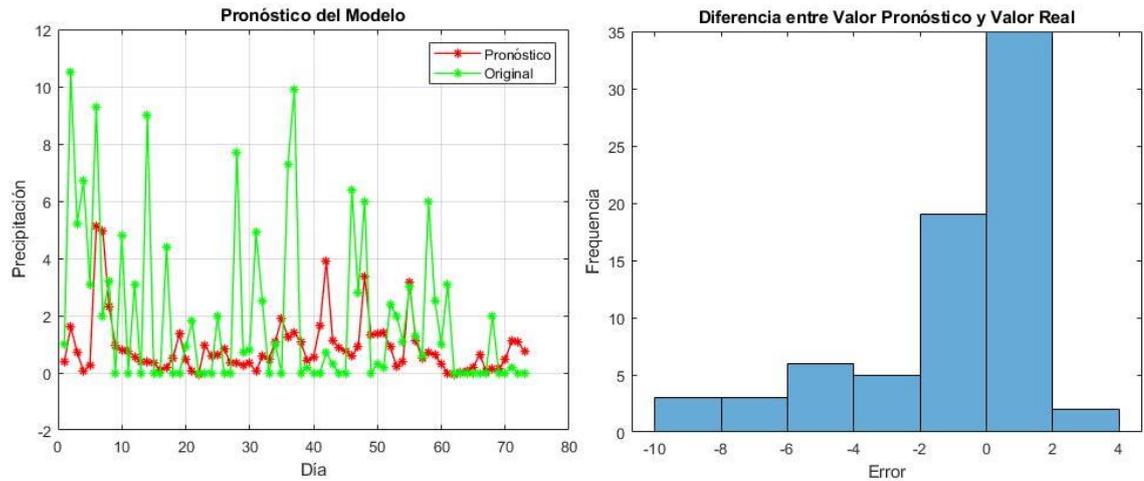
4.2.4 Variación 4

Tabla 12 Variación 4 aplicada a la arquitectura del Modelo 2

Capa / Parámetro	Original	Variación
Factor de caída de tasa de aprendizaje	0,2	1
Tasa de aprendizaje inicial	0,005	0,02
Máximo de Épocas	50	100
Capa Dropout 1	0,2	0,5
Error RMSE	2,6336	2,9096

Fuente: Elaboración propia

Figura 21 Pronóstico generado por la variación 4 del modelo 2 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

El modelo 2 destaca por tener dos funciones de activación Relu, con sus capas completamente conectadas de 100 y 1000 neuronas respectivamente. La capa inicial LSTM tiene 100 unidades ocultas. Además, el error RMSE generado es de 2,6336 siendo relativamente bajo. La gráfica de pronóstico muestra dos picos cercanos a 4 y a 6 mm, sin embargo, no existen gradientes pronunciados como en el comportamiento original. Las variaciones efectuadas en este modelo se hicieron principalmente en el factor de caída de tasa de aprendizaje, tasa de aprendizaje inicial y en el máximo de épocas. Se aumentó el factor de caída de aprendizaje a 1, lo cual permitió tener picos más altos. En la variación 2 se aumentó a 500 el número de épocas, con lo que se desarrollaron más picos. Cabe mencionar que con la variación de la capa dropout y la tasa de aprendizaje inicial, no se lograron mejoras pronunciadas.

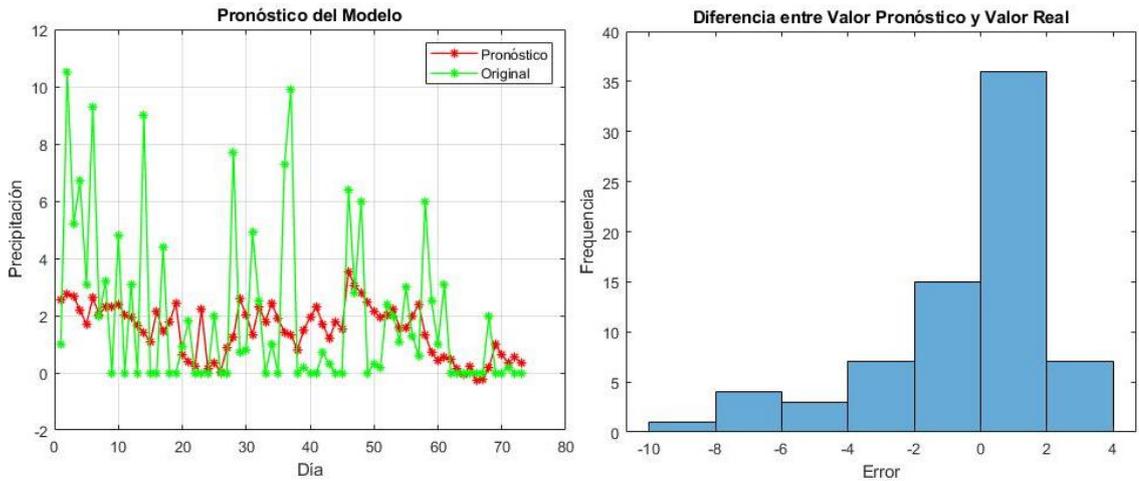
4.3. MODELO 3

Tabla 13 Arquitectura del Modelo 3

Capa / Parámetro	Cantidad de Neuronas / Valor
Unidades Ocultas – Capa LSTM	10
Capa completamente conectada – F. Activación Tanh	500
Capa completamente conectada – F. Activación Tanh	50
Error RMSE	2,6631

Fuente: Elaboración propia

Figura 22 Pronóstico generado por el modelo 3 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

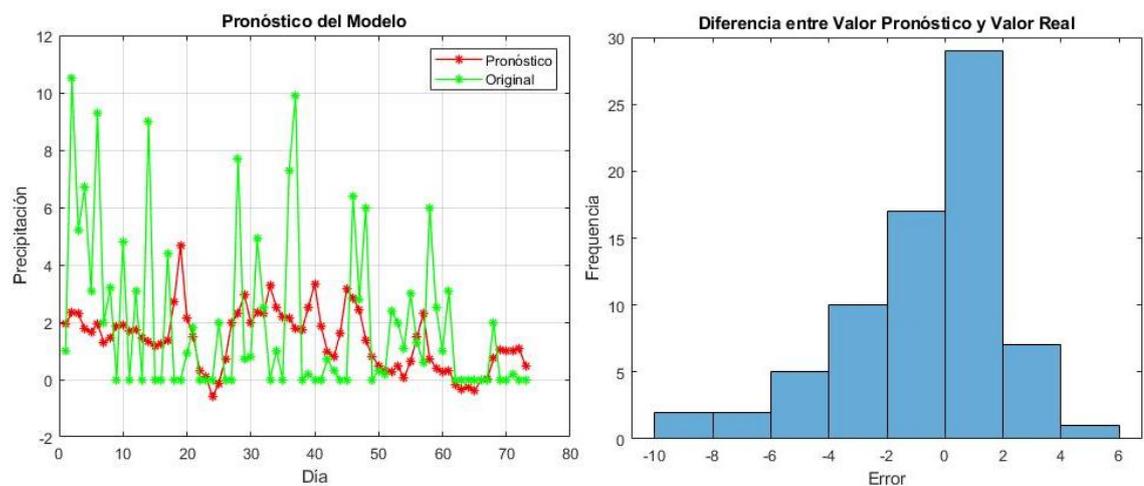
4.3.1. Variación 1

Tabla 14 Variación 1 aplicada a la arquitectura del Modelo 3

Capa / Parámetro	Original	Variación
Taza de aprendizaje inicial	0,005	0,0005
Capas Dropout	Sí	No
Error RMSE	2,6631	2,8082

Fuente: Elaboración propia

Figura 23 Pronóstico generado por la variación 1 del modelo 3 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

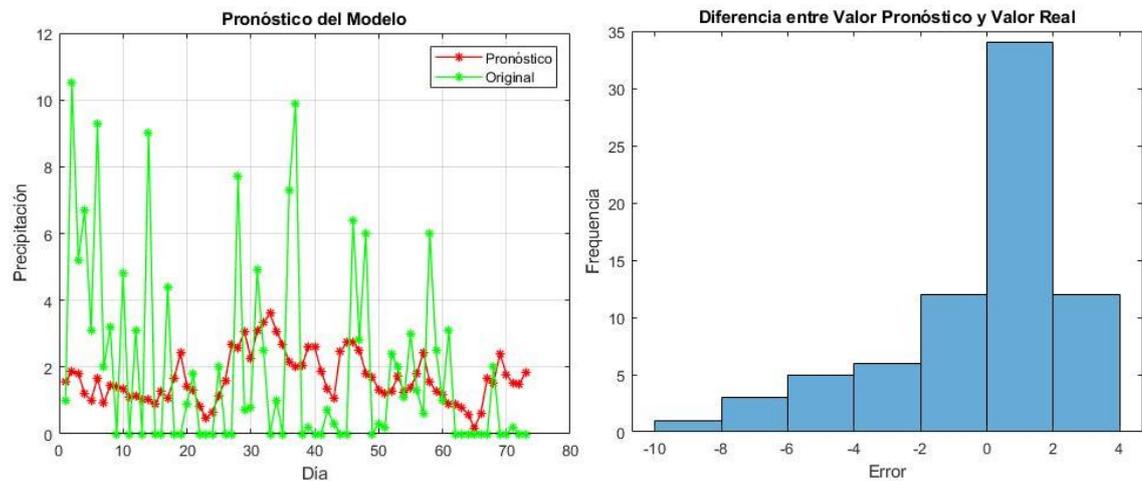
4.3.2. Variación 2

Tabla 15 Variación 2 aplicada a la arquitectura del Modelo 3

Capa / Parámetro	Original	Variación
Taza de aprendizaje inicial	0,005	0,0005
Error RMSE	2,6631	2,8152

Fuente: Elaboración propia

Figura 24 Pronóstico generado por la variación 2 del modelo 3 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

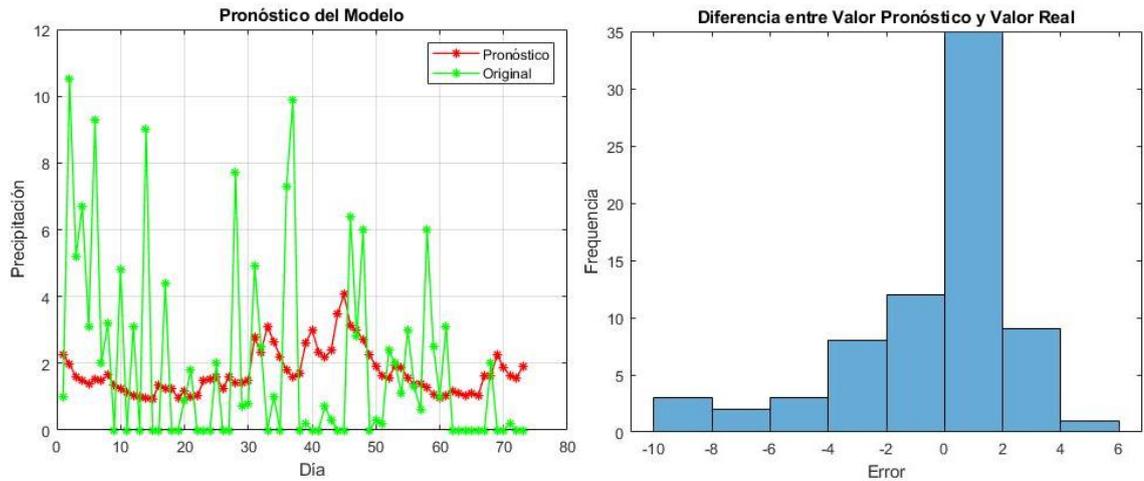
4.3.3 Variación 3

Tabla 16 Variación 3 aplicada a la arquitectura del Modelo 3

Capa / Parámetro	Original	Variación
Taza de aprendizaje inicial	0,005	0,0005
Máximo de Épocas	50	20
Error RMSE	2,6631	2,8150

Fuente: Elaboración propia

Figura 25 Pronóstico generado por la variación 3 del modelo 3 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

El modelo 3 destaca por tener mayor cambio de comportamiento de pendiente de recta, es decir tiene mayor cantidad de picos, sin embargo, los valores no sobrepasan el valor de 4. En la arquitectura de este modelo se colocaron dos funciones de activación Tanh, con sus respectivas capas completamente conectadas de 500 y 50 neuronas respectivamente. Además, la capa LSTM posee 10 unidades ocultas. Las variaciones del modelo se aplicaron principalmente a la tasa de aprendizaje inicial, donde al reducir el valor de 0,005 a 0,0005. Con las modificaciones aplicadas se observaron mejoras en el comportamiento del modelo. Por otro lado, en la primera variación se eliminaron las capas dropout, sin tener una mejora significativa, adicionalmente se redujo el valor de máximo de épocas de 50 a 20. Con lo acotado, no se lograron tener cambios notorios y sobre todo no se aumentaron la cantidad de picos en la gráfica. De las variaciones aplicadas, el cambio en el parámetro de entrenamiento de tasa de aprendizaje inicial es el que más aportó al modelo. Esta última especificación se la tomó como base para las modificaciones en los siguientes modelos.

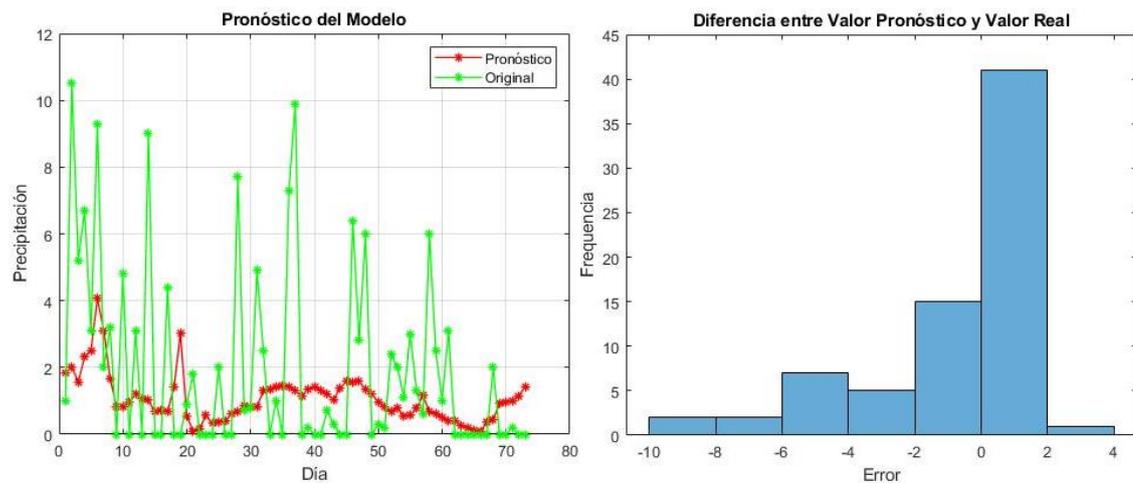
4.4. MODELO 4

Tabla 17 *Arquitectura del Modelo 4*

Capa / Parámetro	Cantidad de Neuronas / Valor
Unidades Ocultas – Capa LSTM	100
Capa completamente conectada – F. Activación Relu	500
Capa completamente conectada – F. Activación Tanh	100
Capa completamente conectada – F. Activación Relu	500
Error RMSE	2,7546

Fuente: Elaboración propia

Figura 26 *Pronóstico generado por el modelo 4 en comparación al original, y Diferencia entre valor pronóstico y valor real*



Fuente: Elaboración propia

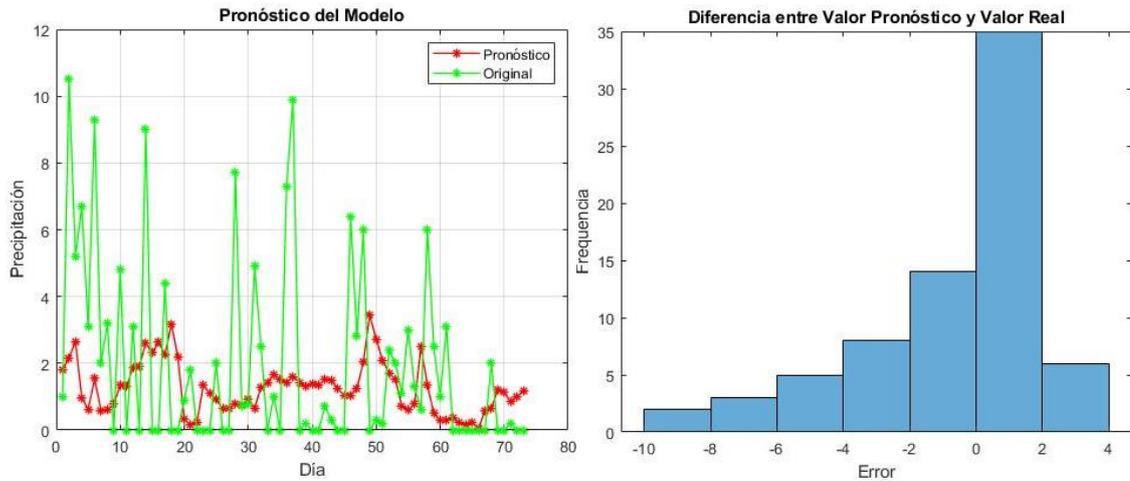
4.4.1 Variación 1

Tabla 18 *Variación 1 aplicada a la arquitectura del Modelo 4*

Capa / Parámetro	Original	Variación
Taza de aprendizaje inicial	0,005	0,0005
Máximo de Épocas	50	200
Error RMSE	2,7546	2,8483

Fuente: Elaboración propia

Figura 27 Pronóstico generado por la variación 1 del modelo 4 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

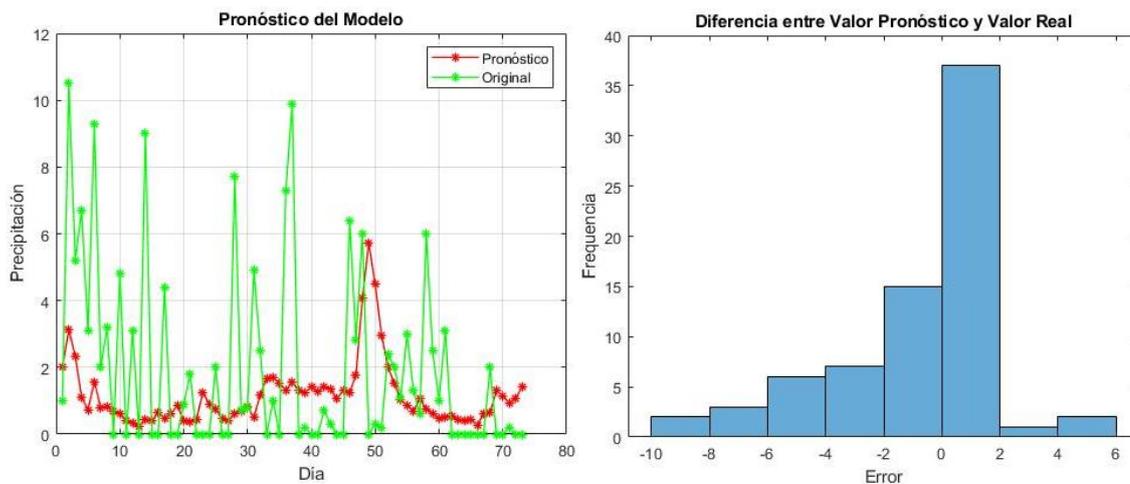
4.4.2. Variación 2

Tabla 19 Variación 2 aplicada a la arquitectura del Modelo 4

Capa / Parámetro	Original	Variación
Taza de aprendizaje inicial	0,005	0,0005
Máximo de Épocas	50	400
Error RMSE	2,7546	2,9299

Fuente: Elaboración propia

Figura 28 Pronóstico generado por la variación 2 del modelo 4 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

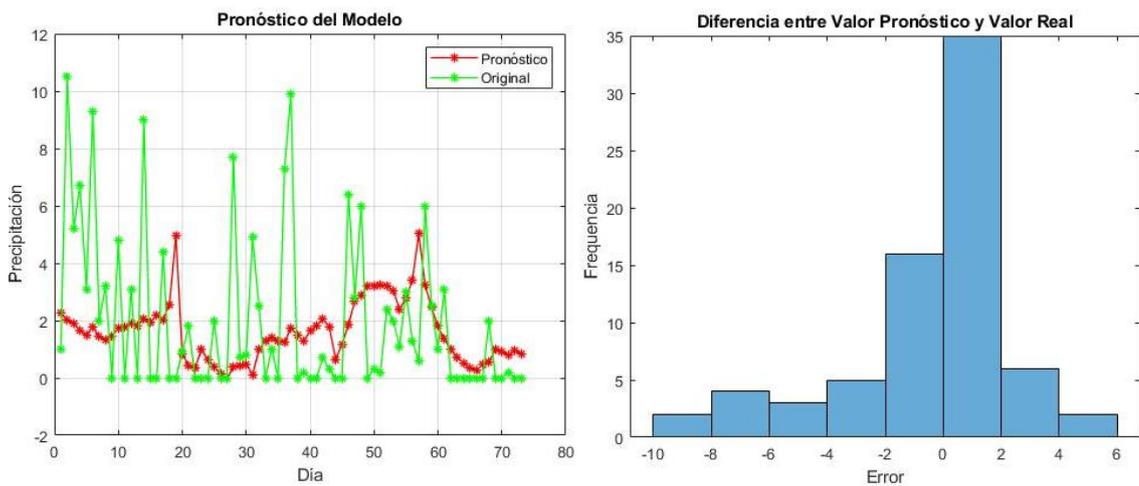
4.4.3. Variación 3

Tabla 20 Variación 3 aplicada a la arquitectura del Modelo 4

Capa / Parámetro	Original	Variación
Taza de aprendizaje inicial	0,005	0,0005
Máximo de Épocas	50	100
Capa Dropout 1	0,2	0,5
Capa Dropout 2	0,2	0
Capa Dropout 3	0,2	0
Capa Dropout 4	0,2	0
Error RMSE	2,7546	2,8636

Fuente: Elaboración propia

Figura 29 Pronóstico generado por la variación 3 del modelo 4 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

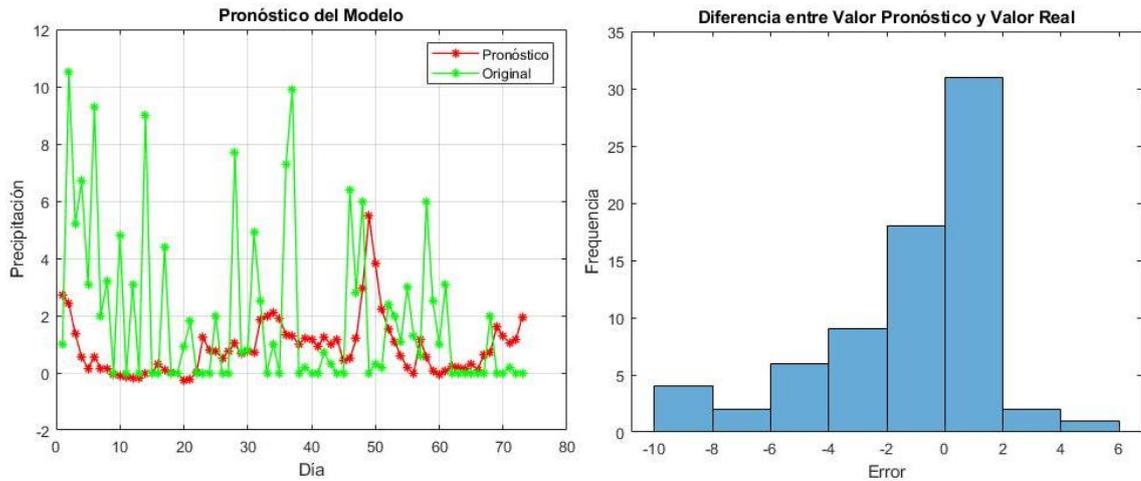
4.4.4. Variación 4

Tabla 21 Variación 4 aplicada a la arquitectura del Modelo 4

Capa / Parámetro	Original	Variación
Taza de aprendizaje inicial	0,005	0,0005
Máximo de Épocas	50	150
Umbral de Gradiente	1	20
Capa Dropout 2	0,2	0
Capa Dropout 3	0,2	0
Capa Dropout 4	0,2	0
Error RMSE	2,7546	3,1267

Fuente: Elaboración propia

Figura 30 Pronóstico generado por la variación 4 del modelo 4 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

El modelo 4 se diferencia de los anteriores, debido a que tiene en el bloque CC + FA + D, 3 funciones de activación con sus respectivas capas completamente conectadas. La primera función de activación es Relu, la cual está acompañada de su respectiva capa completamente conectada con 500 neuronas. La segunda función es Tanh con 100 neuronas en su capa completamente conectada. La última función es tanh, y con 500 neuronas en su CC. Las variaciones aplicadas en este modelo se fundamentan principalmente en la cantidad de máximo de épocas. Se observa un mejor comportamiento del modelo al aumentar el valor. Sin embargo, en un valor extremadamente alto, el modelo empieza a comportarse como una recta con pendiente 0, es decir solo fluctúa en el valor de la media. Por otro lado, se modificó el umbral de gradiente sin tener una mejoría notoria, obteniendo pocos picos sin la presencia de altos errores de RMSE.

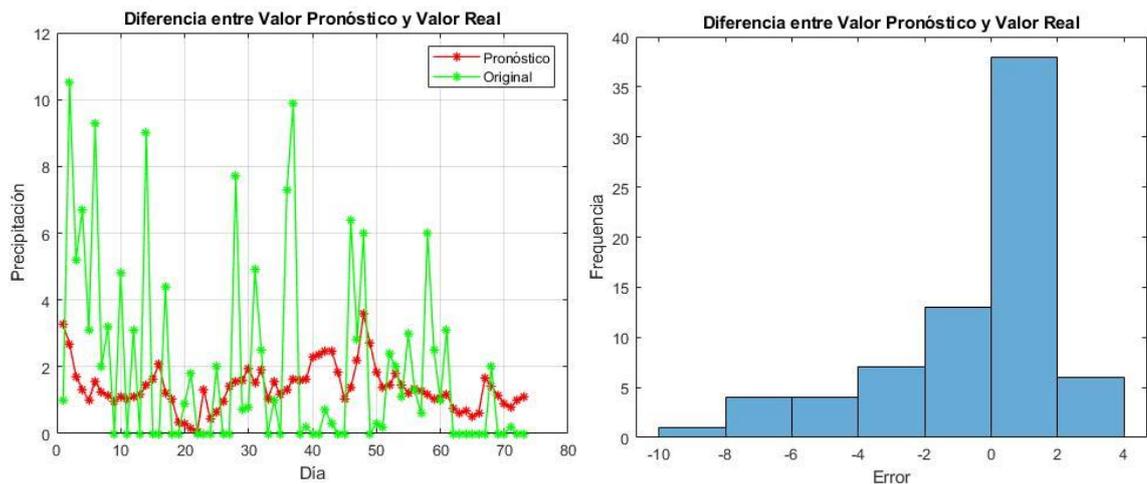
4.5. MODELO 5

Tabla 22 *Arquitectura del Modelo 5*

Capa / Parámetro	Cantidad de Neuronas / Valor
Unidades Ocultas – Capa LSTM	10
Capa completamente conectada – F. Activación Tanh	100
Capa completamente conectada – F. Activación Relu	500
Capa completamente conectada – F. Activación Relu	500
Error RMSE	2,7581

Fuente: Elaboración propia

Figura 31 *Pronóstico generado por el modelo 5 en comparación al original, y Diferencia entre valor pronóstico y valor real*



Fuente: Elaboración propia

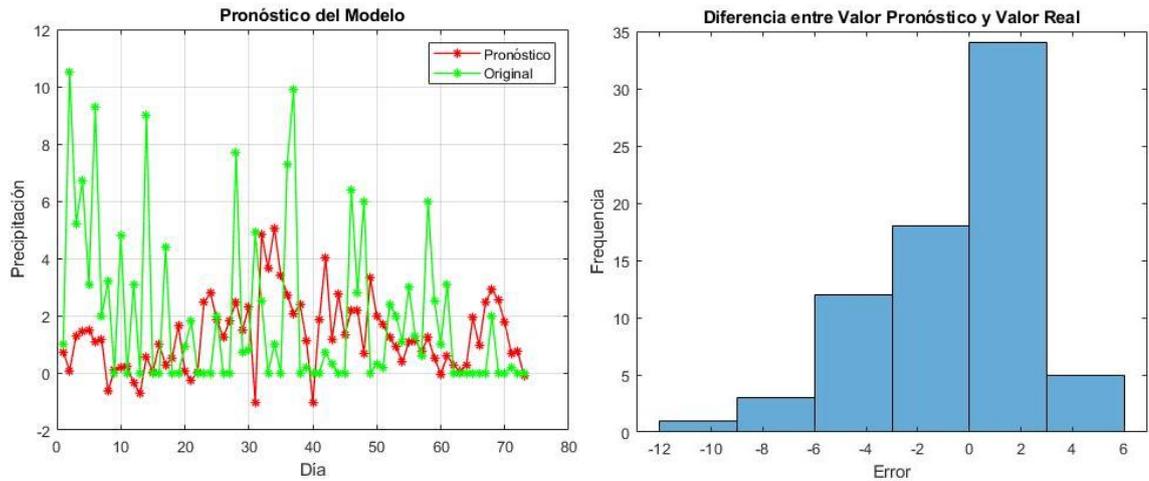
4.5.1. Variación 1

Tabla 23 *Variación 1 aplicada a la arquitectura del Modelo 5*

Capa / Parámetro	Original	Variación
Taza de aprendizaje inicial	0,005	0,0005
Máximo de Épocas	50	150
Capa Dropout 1	0,2	0,01
Capa Dropout 2	0,2	0,02
Capa Dropout 3	0,2	0,02
Capa Dropout 4	0,2	0,02
Error RMSE	2,7581	3,1821

Fuente: Elaboración propia

Figura 32 Pronóstico generado por la variación 1 del modelo 5 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

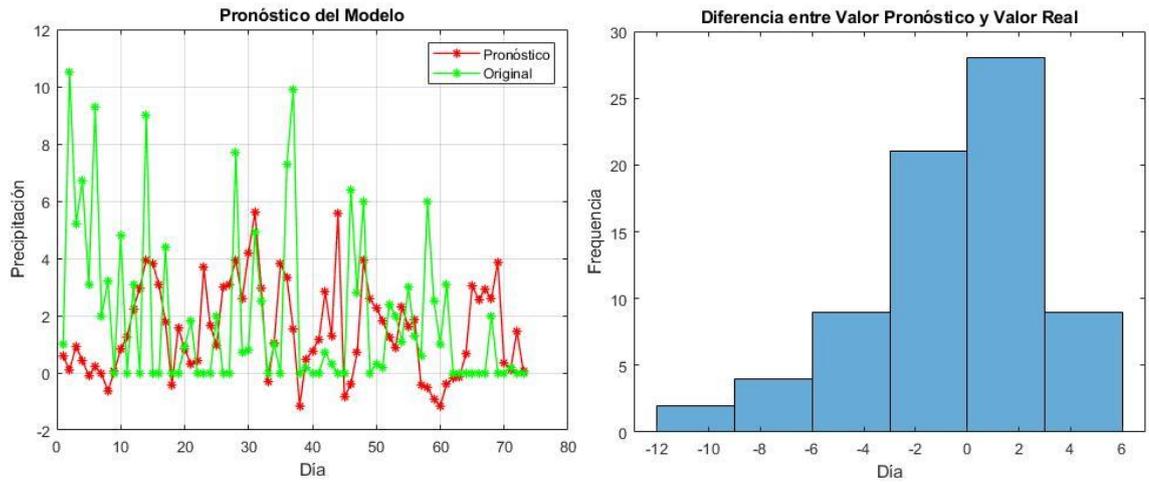
4.5.2. Variación 2

Tabla 24 Variación 2 aplicada a la arquitectura del Modelo 5

Capa / Parámetro	Original	Variación
Capa LSTM	10	5
Taza de aprendizaje inicial	0,005	0,0005
Máximo de Épocas	50	150
Capa Dropout 1	0,2	0,01
Capa Dropout 2	0,2	0,02
Capa Dropout 3	0,2	0,02
Capa Dropout 4	0,2	0,02
Error RMSE	2,7581	3,2302

Fuente: Elaboración propia

Figura 33 Pronóstico generado por la variación 2 del modelo 5 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

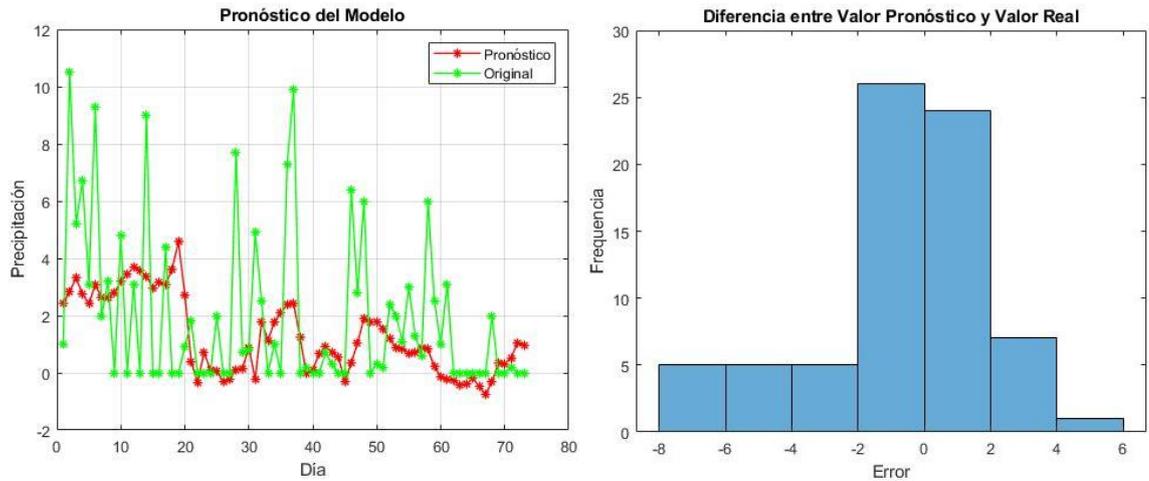
4.5.3. Variación 3

Tabla 25 Variación 3 aplicada a la arquitectura del Modelo 5

Capa / Parámetro	Original	Variación
Capa LSTM	10	500
Taza de aprendizaje inicial	0,005	0,0005
Máximo de Épocas	50	150
Capa Dropout 1	0,2	0,01
Capa Dropout 2	0,2	0,02
Capa Dropout 3	0,2	0,02
Capa Dropout 4	0,2	0,02
Error RMSE	2,7581	2,747

Fuente: Elaboración propia

Figura 34 Pronóstico generado por la variación 3 del modelo 5 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

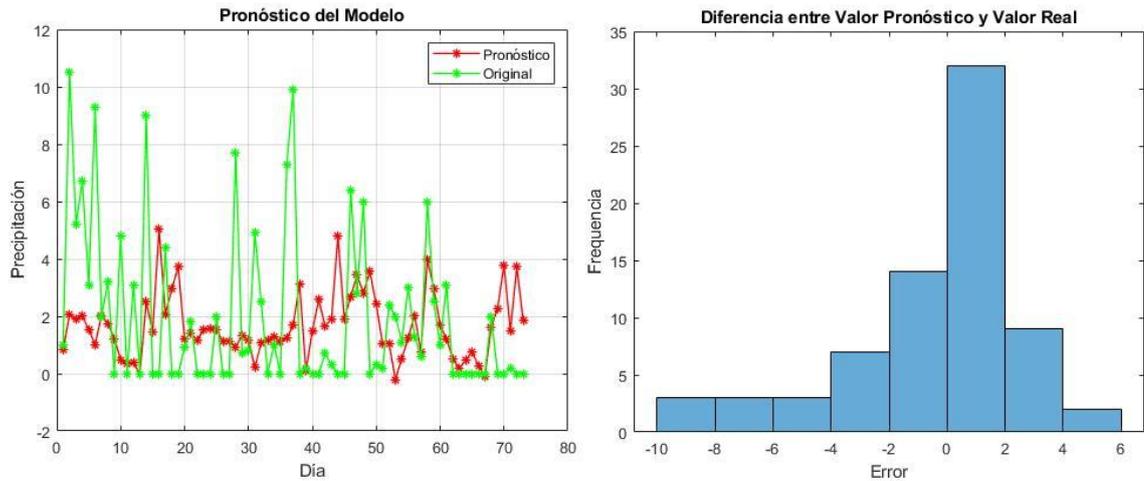
4.5.4. Variación 4

Tabla 26 Variación 4 aplicada a la arquitectura del Modelo 5

Capa / Parámetro	Original	Variación
Capa LSTM	10	2
Taza de aprendizaje inicial	0,005	0,0005
Máximo de Épocas	50	150
Capa Dropout 1	0,2	0,01
Capa Dropout 2	0,2	0,02
Capa Dropout 3	0,2	0,02
Capa Dropout 4	0,2	0,02
Error RMSE	2,7581	2,9592

Fuente: Elaboración propia

Figura 35 Pronóstico generado por la variación 4 del modelo 5 en comparación al original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

El modelo 5 destaca por tener 1 función de activación Relu con su capa completamente conectada de 100 neuronas y posteriormente 2 Relu con sus capas completamente conectadas de 500 neuronas cada una. La capa inicial LSTM tiene 10 unidades ocultas. El comportamiento de este modelo es interesante al tener gradientes pronunciados y con picos de 0 y 4 en precipitación. Las variaciones que se aplicaron van desde modificaciones al factor dropout, bajándolo a niveles menores a 0,1 hasta modificaciones en los parámetros de entrenamiento. Con respecto a lo mencionado, la tasa de aprendizaje inicial, se bajó de 0,005 a 0,0005. En el máximo de épocas se colocó un número de 150. Con todas estas modificaciones se observa una mejora en el modelo, sin embargo, se procedió a modificar las unidades ocultas de LSTM, el cuál era un parámetro fijo no cambiado. Con esta última variación, se obtuvieron mejoras significativas, sobre todo en la gráfica comparativa. El valor que se colocó en unidades ocultas de LSTM, es el mínimo, es decir 2, con lo que la gráfica presentó mayores picos y se acercó más al comportamiento original a pesar de aumentar su error RMSE.

4.6. MODELO FINAL

Después de haber realizado las modificaciones respectivas a los 5 modelos, se escogió la mejor variación. El modelo 5 con su variación 4 fue elegido. Adicionalmente, con cambios puntuales adicionales en los parámetros de entrenamiento, se aplicó la validación final con los datos de testeo. A continuación, se muestran resultados obtenidos y las especificaciones del modelo final.

Tabla 27 *Arquitectura del Modelo Final*

Capa / Parámetro	Cantidad de Neuronas / Valor
Unidades Ocultas – Capa LSTM	2
Capa completamente conectada – F. Activación Tanh	100
Capa completamente conectada – F. Activación Relu	500
Capa completamente conectada – F. Activación Relu	500
Capa Dropout 1	0,01
Capa Dropout 2	0,02
Capa Dropout 3	0,02
Capa Dropout 4	0,02
Error RMSE	4,4535

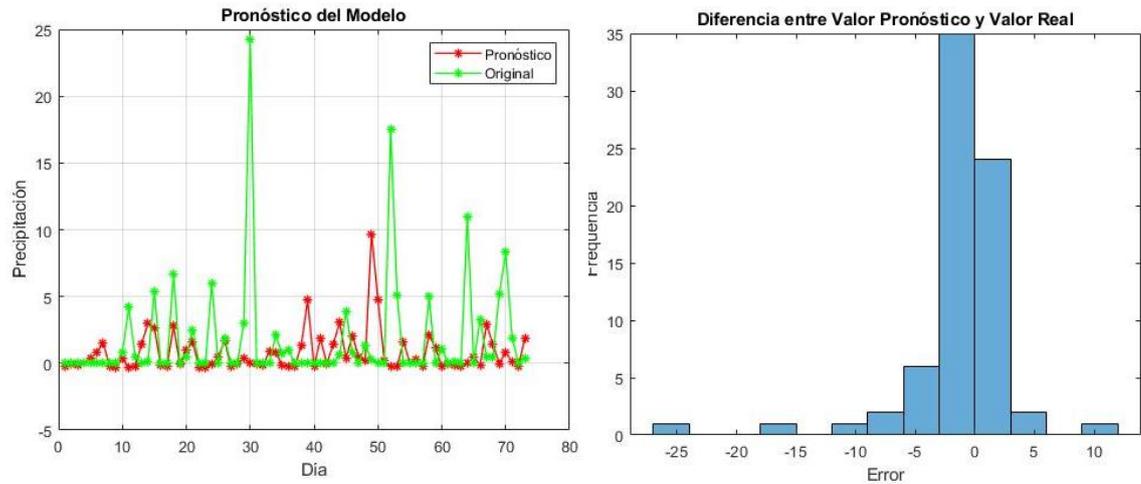
Fuente: Elaboración propia

Tabla 28 *Parámetros de entrenamiento del Modelo Final*

Parámetro de entrenamiento	Valor
Optimizador	Rmsprop
Máximo de Épocas	150
Taza de aprendizaje inicial	0,005
Umbral de Gradiente	1
Período de caída de tasa de aprendizaje	125
Factor de caída de tasa de aprendizaje	0,2

Fuente: Elaboración propia

Figura 36 Pronóstico generado por el modelo Final en comparación al comportamiento original, y Diferencia entre valor pronóstico y valor real



Fuente: Elaboración propia

En el modelo final están presentes todas las especificaciones analizadas en el transcurso de la experimentación. Destaca una capa inicial LSTM con 2 unidades ocultas, debido a que el modelo ofrece una mejor adaptación con pocas unidades ocultas por la cantidad de datos con los que se trabajó. Por otro lado, se utilizó 3 bloques CC+FA+D, con tanh, tanh y relu respectivamente. En los mencionados, se colocaron 100, 500 y 500 neuronas en las capas completamente conectadas de las funciones de activación. El factor dropout utilizado es 0,01 para LSTM, y 0,02 para los bloques CC+FA+D. Dentro de los parámetros de entrenamiento que destacan están: el optimizador utilizado que fue el RMSPROP, mientras que el máximo de épocas fue de 150. El contraste entre unidades ocultas de LSTM y la cantidad de neuronas en el bloque CC+FA+D, muestra mejor adaptación del pronóstico a la gráfica original. El error RMSE incrementó en su valor, superando el valor de 4, en contraste a lo mostrado en la etapa de validación, donde los valores no pasaban del umbral de 3,5. El error RMSE no es un indicador tan preciso del comportamiento de la red neuronal, en el caso de pronóstico de precipitación. Esto debido a que los valores son muy irregulares y el patrón de comportamiento es muy variable. La gráfica viene siendo un mejor patrón visual de análisis de efectividad del modelo que se aplica. En comparación con otros trabajos realizados en la misma área, como por ejemplo a Kp forecasting with a recurrent neural network (Sexton & Ma, 2019), la adaptabilidad del modelo es aceptable. En otras latitudes la precipitación muestra mayor regularidad con

respecto a la estación, sin embargo en la sierra ecuatoriana existen periodos irregulares de lluvia. Por lo mencionado, es comprensible que los resultados no muestren 100% de similitud de la gráfica de pronóstico y la gráfica con valores reales. Otros trabajos muestran gráficas de pronósticos mejores, en función de la cantidad de datos obtenidos para el análisis, además de añadir parámetros adicionales como velocidad del viento. Sin embargo, el presente trabajo, no se aleja de las acotaciones dadas por los autores con respecto a la implementación de redes neuronales recurrentes al clima y tiempo.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Se desarrolló una aplicación de machine learning para el pronóstico de precipitación, a través de redes neuronales artificiales recurrentes usando la celda LSTM. Se tomó como referencia los datos de 7 años de precipitación de la estación meteorológica de Querochacha. Se obtuvo un conjunto de datos de que van desde el 2015 al 2021, y se colocó exitosamente en el formato de lectura de redes neuronales artificiales. La temporada con mayor cantidad de precipitación en el contexto analizado del presente trabajo es la de Junio a agosto, y la de menos precipitación es la de diciembre – enero.
- El número de capas en la arquitectura de la red neuronal utilizada en este proyecto es de 6. La primera capa es la celda LSTM, donde el número de unidades ocultas es de 2. Adicionalmente se colocaron 3 bloques con capa completamente conectada seguida de función de Activación y acompañada con una capa dropout. El primer bloque contiene 100 neuronas con la función de activación tanh, y un dropout con un factor de 0,02. En el segundo bloque se colocaron 500 neuronas acompañadas de la función de activación relu, y con un dropout de 0,02. El tercer bloque tiene 500 neuronas con la función de activación relu, y un dropout de factor 0,02. Finalmente, la capa de salida tiene 1 neurona con función de activación purelin. Esta arquitectura se definió una vez que se realizaron 348 pruebas de experimentación, y haber aplicado variaciones adicionales a los 5 mejores modelos. La experimentación se basó en modificaciones en la cantidad de unidades ocultas de la celda LSTM, y en las neuronas de los bloques de capa completamente conectada con función de activación y dropout.
- Los parámetros de entrenamiento ayudan a mejorar el desempeño del modelo. Específicamente para este trabajo, el máximo de épocas y la tasa de aprendizaje inicial aumentan la adaptabilidad del modelo.

- Con los datos de este proyecto, al colocar una gran cantidad de unidades ocultas en la celda LSTM, no hay una buena predicción, ya que la curva de predicción tiende a ser una constante sin picos de variabilidad.
- En este proyecto, una red neuronal con una cantidad baja de neuronas en las capas completamente conectadas no genera un pronóstico cercano a la realidad, dado que crea curvas de predicción planas.
- Una cantidad excesiva de neuronas en las capas completamente conectadas de los bloques de la red neuronal genera desfase de la gráfica de pronóstico con respecto al comportamiento real de la precipitación.
- En este trabajo se usaron dos tipos de validación, una numérica (RMSE) y una visual que corresponde a la gráfica de predicción versus los datos reales. La métrica RMSE no es suficiente para evaluar el comportamiento de los algoritmos. En casos donde se obtuvo un bajo error, la curva de predicción distaba de forma apreciable a la curva real. Por la razón indicada, se realizó una validación visual de los resultados a través de la curva de pronóstico de predicción de la red como complemento al indicador RMSE.

5.2. Recomendaciones

- Experimentar con modelos de clasificación Booleana aplicados a precipitación, donde el objetivo es predecir la existencia o no de lluvia en un determinado día.
- Utilizar mayor cantidad de datos diarios de precipitación, en periodos de al menos 10 años.
- Realizar una aplicación de redes neuronales recurrentes con datos de precipitación medidos por horas.
- Utilizar otros predictores como velocidad del viento y temperatura para mejorar la predicción de precipitación.
- Aplicar el modelo a otros parámetros meteorológicos como la radiación solar, donde hay mayor cantidad de aplicaciones en la investigación a nivel mundial.
- Reproducir el modelo desarrollado en la presente tesis con datos de otras estaciones meteorológicas del país, e incluso de otras latitudes.

BIBLIOGRAFÍA

- Acevedo, E., Serna, A., & Serna, E. (2018). Principios y características de las redes neuronales artificiales. *Desarrollo e innovación en ingeniería*, 173. Obtenido de https://www.researchgate.net/profile/Jhon-Fredy-Narvaez/publication/320170890_Desarrollos_de_la_Ingenieria_ambiental_en_la_evaluacion_de_la_calidad_de_los_recursos_naturales_y_la_salud_ambiental/links/59d26bfca6fdcc181ad611ce/Desarrollos-de-la-Ingenieria-
- Arana, C. (2021). Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales. *Serie Documentos de Trabajo*(797). Obtenido de <https://www.econstor.eu/bitstream/10419/238422/1/797.pdf>
- Beale, M. H., Hagan, M., & Demuth, H. (2020). *Deep Learning Toolbox User's Guide*. Obtenido de https://ge0mlib.com/papers/Books/04_Deep_Learning_Toolbox_Users_Guide.pdf
- Betolaza, X. M. (2021). *Implementación de redes neuronales en plataformas hardware para su aplicación en ingeniería eléctrica*. Obtenido de https://addi.ehu.es/bitstream/handle/10810/54026/TFM_XabierMaestreBetolaza.pdf?sequence=1&isAllowed=y
- Calvo, D. (7 de diciembre de 2018). *Función de activación – Redes neuronales*. Valladolid: Universidad de Valladolid. Obtenido de <https://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/>
- Chacón, W., ValerianoValdez, K., Umasi, J. y Torres, J. (2016). Predicción de caudales medios diarios en la cuenca del Amazonas aplicando redes neuronales artificiales y el modelo neurodifuso ANFIS. *Research in Computing Science*, 113, 23–35. Obtenido de https://rcs.cic.ipn.mx/2016_113/Prediccion%20de%20caudales%20medios%20diarios%20en%20la%20cuenca%20del%20Amazonas%20aplicando%20redes%20neuronales.pdf
- Cheje, R. W. (2019). *MatLab & Redes Neuronales*. Obtenido de <http://www.revistasbolivianas.ciencia.bo/pdf/rits/n1/n1a24.pdf>

- Ederra, F. L. (2019). *Estimació de la producció d'energia solar/consum emprant xarxes neuronals*. Tesis de posgrado. Obtenido de <https://core.ac.uk/download/pdf/268898942.pdf>
- Fernández, C. C. (2018). *Manual básico de Matlab*. Obtenido de <https://webs.ucm.es/centros/cont/descargas/documento11541.pdf>
- Fierro, A. A. (2020). *Predicción de Series Temporales con Redes*. Argentina: Universidad Nacional de La Plata.
- Guerrero, J. P. (2020). *Redes Recurrentes*. Trabajo de fin de grado. Obtenido de <https://idus.us.es/bitstream/handle/11441/115230/TFG%20DGM%20P%20C3%A9rez%20Guerrero%20Jes%C3%BA.pdf?sequence=1&isAllowed=y>
- Habi, H. V., & Messer, H. (2020). Recurrent Neural Network for Rain Estimation Using Commercial Microwave Links. *IEEE Transactions on Geoscience and Remote Sensing*, 59(5). Obtenido de <https://ieeexplore.ieee.org/abstract/document/9153027>
- Janalta Interactive . (17 de enero de 2022). *www.techopedia.com*. Obtenido de techopedia: <https://www.techopedia.com/definition/33181/training-data>
- Jiménez-Carrión, Gutiérrez-Segura y Celi-Pinzón. (2018). Modelado y Predicción del Fenómeno El Niño en Piura, Perú mediante Redes Neuronales Artificiales usando Matlab. *Información tecnológica*, 29(4). Obtenido de https://www.scielo.cl/scielo.php?pid=S0718-07642018000400303&script=sci_arttext
- Li, Liu, Liu y Zhang. (2017). Predictive Power of Machine Learning for Optimizing Solar Water Heater Performance: The Potential Application of High-Throughput Screening. *International Journal of Photoenergy*, 1–10. doi:10.1155/2017/4194251
- Makridakis, Spiliotis y Assimakopoulos. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PloS uno*, 13(3). Obtenido de <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0194889>

- MathWorks. (2022). *MathWorks*. Obtenido de <https://la.mathworks.com/products/deep-learning.html>
- Menacho, C. (2014). *Modelos de regresión lineal con redes neuronales*. Lima: Universidad Nacional Agraria La Molina.
- Mohd, M. R., Johari, J., & Ruslán, F. A. (2020). Application of NNARX in Modeling a Solar Radiation Prediction. *Conferencia IEEE sobre Sistemas, Procesos y Control (ICSPC)*. Obtenido de <https://ieeexplore.ieee.org/abstract/document/9305787>
- Olabe, X. B. (2008). *Redes neuronales artificiales y sus aplicaciones*. Bilbao: Escuela Superior de Ingeniería de Bilbao, EHU.
- Ordoñez-Palacios, León-Vargas, Bucheli-Guerrero y Ordoñez-Eraso. (2020). Predicción de radiación solar en sistemas fotovoltaicos utilizando técnicas de aprendizaje automático. *Revista Facultad de Ingeniería*, 29(54). Obtenido de http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-11292020000100031
- Pontigo, S. E. (2020). *Aplicación de redes neuronales recurrentes y modelos de series de tiempo bayesianos a la predicción de rentabilidad de fondos de pensiones*. Tesis de grado. Obtenido de <http://repositorio.udec.cl/jspui/bitstream/11594/526/1/Tesis%20Aplicaci%C3%B3n%20de%20Redes%20Neuronales%20Recurrentes%20y%20Modelos%20de%20Series%20de%20Tiempo%20Image.Marked.pdf>
- Reboita, Gan, Rocha, y Ambrizzi. (2010). Regimes de precipitao na América do Sul: Uma revisado bibliográfica. *Revista Brasileira Meteorológica*(25), 185-204. Obtenido de <https://www.scielo.br/j/rbmet/a/yhrG5QPXhBckZXcGKgyzWvf/?lang=pt>
- Recuero de los Santos P. (24 de enero de 2022). *Datos de entrenamiento vs datos de test*. España: Telefónica Tech AI of Things. Obtenido de <https://empresas.blogthinkbig.com/datos-entrenamiento-vs-datos-de-test/>
- Red Hidrometeorológica de Tungurahua . (10 de 2022). *tungurahua.gob.ec/red/estaciones*. Obtenido de rrnn.tungurahua.gob.ec:

<https://rrnn.tungurahua.gob.ec/red/estaciones/estacion/530b84ed74daaf23bce53ced>

- Rivas, W., & Mazón, B. (2018). *Redes neuronales artificiales aplicadas al reconocimiento de patrones*. Editorial UTMACH. Obtenido de https://www.researchgate.net/profile/Bertha-Mazon-Olivo/publication/327703478_Capitulo_1_Generalidades_de_las_redes_neuronales_artificiales/links/5b9fe3c0299bf13e6038a1d8/Capitulo-1-Generalidades-de-las-redes-neuronales-artificiales.pdf
- Rueda, F. H. (2020). *Clasificación de registros ecg de corta duración usando redes neuronales recurrentes*. Tesis de Maestría. Obtenido de https://www.researchgate.net/profile/Fabio-Hernandez/publication/343944212_Clasificacion_de_registros_ECG_de_corta_duracion_usando_redes_neuronales_recurrentes/links/5f490cb8458515a88b7ce196/Clasificacion-de-registros-ECG-de-corta-duracion-usando-redes-ne
- Salas, J. S. (2018). *Desarrollo con matlab de una red neuronal para estimar la demanda de energía eléctrica*. Trabajo de fin de master . Obtenido de <https://core.ac.uk/download/pdf/211101638.pdf>
- Sarasa-Cabezuelo, A. (2022). Prediction of Rainfall in Australia Using Machine Learning. *13*(163). Obtenido de <https://www.mdpi.com/2078-2489/13/4/163/htm>
- Scher, S., & Messori, G. (2018). Predicting Weather Forecast Uncertainty with Machine Learning. *RMets*. Obtenido de <https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3410>
- Sexton, E. S., & Ma, K. N. (2019). Pronóstico de Kp con una red neuronal recurrente. *J. Space Weather Space Clim.*, *9*(A19). Obtenido de https://www.swsc-journal.org/articles/swsc/full_html/2019/01/swsc180037/swsc180037.html
- Velasquez, J. (2011). *Acotación del error de modelos de redes neuronales Aplicados al pronóstico de series de tiempo*. Bogota: Universidad Nacional de Colombia.
- Vuille, Bradley, y Keimig. (2000). Climate variability in the Andes of Ecuador and its relation to tropical Pacific and Atlantic sea surface temperature anomalies.

Diario del clima, 2520-2535. Obtenido de https://journals.ametsoc.org/view/journals/clim/13/14/1520-0442_2000_013_2520_cvitao_2.0.co_2.xml

Wang, Cai, Zhou, Aziz, Qin, Voropai y Barakhtenko. (2020). Solar irradiance forecasting based on direct explainable neural network. *Energy Conversion and Management*, 226. doi:doi.org/10.1016/j.enconman.2020.113487

Zipparro y Morellato. (2004). Climate as a proximate factor affecting phenology of Atlantic rain forest trees. *AFFAPUBIO*.

ANEXOS

Anexo 1 RMSE obtenido en experimentación

Tabla 29 *RMSE en experimentación con FA tanh*

BLOQUE CC+FA+D FA: TANH		
Unidades Ocultas en LSTM		
Neuronas en CC	10	100
10	2,9173	3,2219
50	2,8201	3,0562
100	3,0305	3,0385
500	2,9438	3,0264
1000	2,818	3,6047

Fuente: Elaboración propia

Tabla 30 *RMSE en experimentación con FA relu*

BLOQUE CC+FA+D FA: RELU		
Unidades Ocultas en LSTM		
Neuronas en CC	10	100
10	2,811	3,3613
50	2,8565	2,8061
100	2,9277	3,1191
500	2,9046	2,7123
1000	2,8949	2,7957

Fuente: Elaboración propia

Tabla 31 RMSE en experimentación con 2 Bloques CC+FA+D, FA: Tanh y Relu

FA: Tanh	FA: Relu	Unidades Ocultas en LSTM		
		Neuronas en CC		
			10	100
10	10		2,8053	2,7758
10	50		2,9864	2,8935
10	100		2,9516	3,1416
10	500		2,8295	2,7617
10	1000		2,943	6,7741
50	10		3,2285	3,2285
50	50		2,8831	3,3161
50	100		2,9052	2,9606
50	500		2,7225	2,88
50	1000		2,9582	3,3684
100	10		2,7093	2,818
100	50		2,7749	2,9996
100	100		2,9687	2,8026
100	500		2,8291	2,63
100	1000		2,9455	2,9355
500	10		2,8812	3,1992
500	50		2,8619	2,6697
500	100		2,9553	2,7397
500	500		2,8225	2,8665
500	1000		2,9737	3,0553
1000	10		2,9026	2,7653
1000	50		2,9583	4,1094
1000	100		2,8126	3,2009
1000	500		3,2404	3,2012
1000	1000		3,2082	2,7977

Fuente: Elaboración propia

Tabla 32 *RMSE en experimentación con 2 Bloques CC+FA+D, FA: Relu y Tanh*

FA: Relu	FA: Tanh	Unidades Ocultas en LSTM	
Neuronas en CC		10	100
10	10	2,7572	2,8538
10	50	2,8289	3,0201
10	100	2,8115	2,9982
10	500	2,8387	2,6973
10	1000	2,7846	3,1095
50	10	2,8337	2,9853
50	50	2,9982	3,2594
50	100	2,9922	2,7436
50	500	2,806	3,2289
50	1000	2,8354	2,9296
100	10	3,1933	2,8446
100	50	2,8443	2,7821
100	100	2,9392	2,8537
100	500	2,9522	2,9882
100	1000	2,7614	3,1118
500	10	3,1704	3,0358
500	50	3,2479	3,1006
500	100	3,0779	3,1012
500	500	2,8872	3,2832
500	1000	3,6094	3,6043
1000	10	2,9727	2,9713
1000	50	3,1071	2,9599
1000	100	2,9811	2,88
1000	500	2,7714	3,8622
1000	1000	2,9867	3,8337

Fuente: Elaboración propia

Tabla 33 RMSE en experimentación con 2 Bloques CC+FA+D, FA: Relu y Relu

FA: Relu	FA: Relu	Unidades Ocultas en LSTM	
		10	100
10	10	2,9684	2,8789
10	50	2,9185	2,9211
10	100	2,8031	2,7908
10	500	3,2524	2,7155
10	1000	2,8695	2,9015
50	10	2,8463	2,8342
50	50	2,86	3,0835
50	100	2,8232	2,8721
50	500	2,8392	3,0752
50	1000	2,8771	2,8728
100	10	2,7517	2,8489
100	50	2,8207	2,9336
100	100	2,7997	3,4195
100	500	2,8919	3,0073
100	1000	2,8342	2,6336
500	10	2,7119	2,8845
500	50	2,9939	3,0277
500	100	2,8842	2,9292
500	500	2,9657	4,7824
500	1000	2,8866	2,99
1000	10	2,839	2,8854
1000	50	2,9668	2,717
1000	100	2,9209	2,9838
1000	500	2,8388	3,6979
1000	1000	2,7543	3,3033

Fuente: Elaboración propia

Tabla 34 RMSE en experimentación con 2 Bloques CC+FA+D, FA: Tanh y Tanh

FA: Tanh	FA: Tanh	Unidades Ocultas en LSTM	
Neuronas en CC		10	100
10	10	2,9531	3,2335
10	50	2,9156	3,067
10	100	2,9651	4,1321
10	500	2,9822	3,0336
10	1000	2,9225	2,9642
50	10	2,9303	3,6867
50	50	3,0964	2,9674
50	100	2,9983	2,9638
50	500	2,9627	3,0014
50	1000	2,9987	3,0295
100	10	3,0196	3,0925
100	50	2,8949	2,9401
100	100	3,0025	2,922
100	500	2,7825	3,3597
100	1000	2,8917	3,0315
500	10	2,9834	3,3674
500	50	2,6631	3,5023
500	100	2,8779	3,32
500	500	3,0476	3,1895
500	1000	2,8175	3,6706
1000	10	2,9823	2,9399
1000	50	3,0615	3,0628
1000	100	3,0259	3,0935
1000	500	2,7227	3,1561
1000	1000	3,2725	3,016

Fuente: Elaboración propia

Tabla 35 RMSE en experimentación con 3 Bloques CC+FA+D, FA: Relu, Relu y Relu

FA: Relu	FA: Relu	FA: Relu	Unidades Ocultas en LSTM	
			Neuronas en CC	
			10	100
100	100	100	2,8699	2,7907
100	100	500	3,0128	3,1305
100	500	100	2,9291	3,4
100	500	500	2,7976	2,7791
500	100	100	2,8923	3,7225
500	100	500	2,9869	3,0425
500	500	100	2,8975	2,9305
500	500	500	3,0554	2,95

Fuente: Elaboración propia

Tabla 36 RMSE en experimentación con 3 Bloques CC+FA+D, FA: Relu, Relu y Tanh

FA: Relu	FA: Relu	FA: Tanh	Unidades Ocultas en LSTM	
			Neuronas en CC	
			10	100
100	100	100	2,9164	2,898
100	100	500	2,8449	2,957
100	500	100	3,0808	2,7834
100	500	500	2,7337	3,5075
500	100	100	2,9538	3,1308
500	100	500	2,9453	3,7366
500	500	100	2,9419	3,0651
500	500	500	2,8876	3,0365

Fuente: Elaboración propia

Tabla 37 RMSE en experimentación con 3 Bloques CC+FA+D, FA: Relu, Tanh y Relu

FA: Relu	FA: Tanh	FA: Relu	Unidades Ocultas en LSTM	
			Neuronas en CC	
			10	100
100	100	100	2,8327	2,8329
100	100	500	2,7745	2,8899
100	500	100	2,9476	2,7773
100	500	500	2,7475	3,5427
500	100	100	2,935	2,8486
500	100	500	2,9041	2,7546
500	500	100	2,7414	2,9162
500	500	500	2,8409	3,4234

Fuente: Elaboración propia

Tabla 38 RMSE en experimentación con 3 Bloques CC+FA+D, FA: Relu, Tanh y Tanh

FA: Relu	FA: Tanh	FA: Tanh	Unidades Ocultas en LSTM	
			Neuronas en CC	
			10	100
100	100	100	3,0353	3,3993
100	100	500	3,0242	3,1221
100	500	100	2,9091	3,2367
100	500	500	2,8493	3,0542
500	100	100	3,1342	3,4372
500	100	500	2,9672	3,0414
500	500	100	2,7692	3,1073
500	500	500	2,8441	3,5535

Fuente: Elaboración propia

Tabla 39 RMSE en experimentación con 3 Bloques CC+FA+D, FA: Tanh, Relu y Relu

FA: Tanh	FA: Relu	FA: Relu	Unidades Ocultas en LSTM	
Neuronas en CC			10	100
100	100	100	2,9397	2,7301
100	100	500	2,8108	2,9624
100	500	100	2,8294	3,1675
100	500	500	2,7581	3,0417
500	100	100	2,8135	2,8665
500	100	500	3,0472	3,0014
500	500	100	3,0396	3,3591
500	500	500	2,9359	2,7911

Fuente: Elaboración propia

Tabla 40 RMSE en experimentación con 3 Bloques CC+FA+D, FA: Tanh, Relu y Tanh

FA: Tanh	FA: Relu	FA: Tanh	Unidades Ocultas en LSTM	
Neuronas en CC			10	100
100	100	100	2,9766	2,8357
100	100	500	2,8564	3,4381
100	500	100	2,8962	2,8526
100	500	500	2,8995	3,0311
500	100	100	2,9185	3,0048
500	100	500	2,7889	3,0549
500	500	100	2,9264	3,2352
500	500	500	2,7478	3,5926

Fuente: Elaboración propia

Tabla 41 RMSE en experimentación con 3 Bloques CC+FA+D, FA: Tanh, Tanh y Relu

FA: Tanh	FA: Tanh	FA: Relu	Unidades Ocultas en LSTM	
			10	100
	Neuronas en CC			
100	100	100	2,8033	3,6651
100	100	500	2,7303	2,778
100	500	100	2,7645	2,9367
100	500	500	3,4441	2,941
500	100	100	2,8568	3,6052
500	100	500	2,8788	3,0021
500	500	100	2,7924	2,9405
500	500	500	3,1501	2,7312

Fuente: Elaboración propia

Tabla 42 RMSE en experimentación con 3 Bloques CC+FA+D, FA: Tanh, Tanh y Tanh

FA: Tanh	FA: Tanh	FA: Tanh	Unidades Ocultas en LSTM	
			10	100
	Neuronas en CC			
100	100	100	2,9402	3,4496
100	100	500	2,8462	2,9427
100	500	100	3,1106	3,0779
100	500	500	2,9122	3,1015
500	100	100	2,7027	2,9284
500	100	500	2,7664	3,1595
500	500	100	2,7708	3,0791
500	500	500	2,7471	3,2637

Fuente: Elaboración propia

Anexo 2 Código de MatLab utilizado en el proyecto

```
clc
clear all
close all

Xtrain = xlsread('datatrain.xlsx',1, 'A1:F219');
Ytrain = xlsread('datatrain.xlsx',1, 'G1:G219');

Xvalidation = xlsread('datavalidation.xlsx',1, 'A1:F73');
Yvalidation = xlsread('datavalidation.xlsx',1, 'G1:G73');

Xtest = xlsread('datatest.xlsx',1, 'A1:F73');
Ytest = xlsread('datatest.xlsx',1, 'G1:G73');

muxtrain = mean(Xtrain,2);
sigxtrain = std(Xtrain,0,2);
XTrainStandardized = (Xtrain - muxtrain)./sigxtrain

muytrain = mean(Ytrain);
sigytrain = std(Ytrain);
YTrainStandardized = (Ytrain - muytrain)/sigytrain

XValidationStandardized = (Xvalidation - muxtrain)./sigxtrain
YValidationStandardized = (Yvalidation - muytrain)/sigytrain

XTestStandardized = (Xtest - muxtrain)./sigxtrain
YTestStandardized = (Ytest - muytrain)/sigytrain

inputSize = 6;
numResponses = 1;
numHiddenUnits = 2;

layers = [
    sequenceInputLayer(inputSize,"Name","sequenceinput")
    lstmLayer(numHiddenUnits,"Name","lstm")
    dropoutLayer(0.01,"Name","dropout1")
    fullyConnectedLayer(100,"Name","fc_1")
    tanhLayer("Name","tanh_1")
    dropoutLayer(0.02,"Name","dropout2")
    fullyConnectedLayer(500,"Name","fc_2")
    reluLayer("Name","relu1")
    dropoutLayer(0.02,"Name","dropout3")
    fullyConnectedLayer(500,"Name","fc_3")
    reluLayer("Name","relu2")
    dropoutLayer(0.02,"Name","dropout4")
    fullyConnectedLayer(1,"Name","fc_4")
    regressionLayer("Name","regressionoutput")];
```

```

options = trainingOptions('rmsprop', ...
    'MaxEpochs',150, ...
    'GradientThreshold',1, ...
    'InitialLearnRate',0.0005, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',125, ...
    'LearnRateDropFactor', 0.2, ...
    'Verbose',0, ...
    'Shuffle','never', ...
    'Plots','training-progress');

net = trainNetwork(XTrainStandardized,YTrainStandardized,layers,options);

YPred1 = predict(net,XTestStandardized)

YPredDes = YPred1*sigytrain + muytrain

figure,
plot(YPredDes,'r-*','LineWidth',1), hold on
plot(Ytest,'g-*','LineWidth',1); hold all;
xlabel('Día')
ylabel('Precipitación')
grid on
legend('Predicción','Original');
title('Pronóstico')

figure
rmse = sqrt(mean((YPredDes - Ytest).^2))
histogram(YPredDes - Ytest)
title("RMSE = " + rmse)
ylabel("Frequency")
xlabel("Error")

```