Máster Universitario en Sistemas Inteligentes
Trabajo de Fin de Máster
Julio, 2023

Benchmarking de controladores inteligentes distribuidos con métodos de aprendizaje profundo

Presentado por:
Geovanny Daniel Satama Bermeo
Dirigido por:
Pastora Isabel Vega Cruz
Mario Francisco Sutil

Departamento de Informática y Automática
Universidad de Salamanca

Agradecimiento

Quiero expresar mi gratitud a la Universidad por brindarme la oportunidad de adquirir conocimientos y desarrollarme en el campo de la inteligencia artificial. Su enfoque en la excelencia académica y los recursos proporcionados fueron fundamentales para mi crecimiento y aprendizaje durante todo el programa.

Mi agradecimiento especial se dirige a la profesora Pastora Vega Cruz, cuya guía y apoyo incondicional han sido invaluables en mi camino hacia la finalización de este trabajo de fin de máster. Su experiencia y dedicación me han inspirado a alcanzar nuevos niveles de excelencia, y estoy verdaderamente agradecido por su tiempo y esfuerzo.

Además, quiero expresar mi sincero agradecimiento a Oscar Aponte Rengifo por su valiosa contribución a mi investigación. Su conocimiento y experiencia en el sistema sobre el que se basó este trabajo fueron fundamentales para su éxito. Estoy agradecido por las discusiones enriquecedoras, las orientaciones y el apoyo que brindó a lo largo de este proceso.

También me gustaría recordar y agradecer a mis seres queridos que ya no están con nosotros, pero que siguen siendo una fuente de inspiración para mí. Quiero expresar mi gratitud a mis abuelitos en el cielo, quienes siempre creyeron en mí y me animaron a seguir adelante en mi educación. Su amor y apoyo incondicional continúan guiándome.

A mis padres, quiero agradecerles por su constante apoyo y aliento a lo largo de mi viaje académico. Su amor incondicional, sabiduría y sacrificio han sido la base de mi éxito. No tengo palabras suficientes para expresar mi gratitud por todo lo que han hecho por mí.

A mi hermana, Carmita, y a Allison, quiero agradecerles por su apoyo inquebrantable y por estar siempre a mi lado durante este desafiante proceso. Su aliento y palabras de ánimo han sido un verdadero impulso para mí, y estoy agradecido por su presencia en mi vida.

Por último, pero no menos importante, quiero agradecer a todos mis amigos y seres queridos que me han apoyado a lo largo de esta etapa. Sus palabras de aliento, gestos amables y comprensión me han dado la fuerza y la motivación necesarias para superar los obstáculos y alcanzar mis metas.

Resumen

En esta investigación, se implementaron las redes neuronales profundas, Multi Layer Perceptron (MLP) y Long-Short Term Memory (LSTM), para optimizar el tratamiento de aguas residuales. Debido a la importancia crucial de asegurar que las EDAR cumplan con los límites legales establecidos para los contaminantes, el modelado y la predicción precisa de estos procesos es vital. Con este propósito, se seleccionaron las redes MLP y LSTM por su capacidad para procesar y aprender de los datos en los sistemas en lazo abierto como en lazo cerrado. Este enfoque permite detectar de manera eficiente las posibles violaciones en la calidad del efluente. En cuanto a los resultados obtenidos, se encontró que las predicciones hechas por la red MLP mostraron una menor tasa de error en comparación con las realizadas por la red LSTM. Sin embargo, se identificaron desafíos durante el entrenamiento de ambas redes, tales como una tendencia al sobreajuste en las iteraciones posteriores. Esto subraya la necesidad de desarrollar estrategias de regularización y optimización de hiperparámetros para mejorar el rendimiento de los modelos de redes neuronales en el tratamiento de aguas residuales.

Abstract

In this research, deep neural networks, specifically Multi-Layer Perceptron (MLP) and Long-Short Term Memory (LSTM), were implemented to optimize wastewater treatment. Given the critical importance of ensuring that wastewater treatment plants comply with established legal limits for contaminants, the modeling and accurate prediction of these processes is vital. For this purpose, MLP and LSTM networks were chosen due to their ability to process and learn from data in both open-loop and closed-loop systems. This approach enables efficient detection of potential violations in effluent quality. Regarding the results obtained, it was found that predictions made by the MLP network exhibited a lower error rate than those made by the LSTM network. However, challenges were identified during the training of both networks, such as a tendency towards overfitting in later iterations. This underscores the need to develop regularization strategies and hyperparameter optimization to improve the performance of neural network models in wastewater treatment.

Índice

Índice de figuras v

Indice de tablas vi

1. Introducción 1
1.1. Objetivos . . . 2
1.1.1. Objetivo general . . . 2
1.1.2. Objetivos específicos . . . 2

2. Estado del arte 3
2.1. Aplicación de controles inteligentes distribuidos . . . 3
2.2. Aprendizaje profundo . . . 4
2.3. Predicción de la calidad del agua con redes neuronales MLP y LSTM 5
2.4. Optimización del simulador BSM1 mediante control predictivo . . . 6

3. Redes neuronales 6
3.1. Redes neuronales artificiales . . . 7
3.1.1. Características de las redes neuronales artificiales . . . 8
3.1.2. Aplicaciones de las redes neuronales artificiales . . . 9
3.1.3. Redes neuronales profundas . . . 10
3.2. Red perceptrón multicapa (MLP) . . . 11
3.2.1. Estructura de la red perceptrón multicapa . . . 11
3.2.2. Proceso de aprendizaje de la red perceptrón multicapa . . . 12
3.3. Long-short term memory (LSTM) . . . 13
3.3.1. Estructura de la red LSTM . . . 13
3.3.2. Proceso de aprendizaje de la red LSTM . . . 14
3.4. Funciones de activación neuronal . . . 15
3.4.1. Activación unidad lineal rectificada (ReLU) . . . 15
3.4.2. Activación sigmoide . . . 16
3.4.3. Activación de tangente hiperbólica . . . 16
3.5. Elementos del algoritmo de propagación hacia atrás . . . 16
3.5.1. Descenso de gradiente (GD) . . . 17
3.5.2. Descenso de gradiente estocástico (SGD) . . . 17
3.6. Parametrización de modelos . . . 18
3.6.1. Número de épocas . . . 18
3.6.2. Tamaño del lote . . . 18
3.6.3. Tasa de aprendizaje . . . 18
3.6.4. Momento . . . 19

3.6.5. Recorte del gradiente 19

4. Descripción y modelo matemático del proceso de fangos activados 20
4.1. Etapas de tratamiento de las aguas residuales 20
4.2. Descripción del Proceso de Fangos Activados 20
4.3. Plataforma de simulación BSM1 21
4.4. Modelo de planta 22
4.5. Perturbaciones en el influente 23

5. Descripción del Caso de Estudio 23
5.1. Obtención de datos de la plataforma de simulación BSM1 24
5.2. Métodos estadísticos propuestos 25
5.3. Análisis de los resultados usando Coeficiente de correlación de Pearson 25
5.3.1. Correlación de Pearson – Punto 1(Influente) 26
5.3.2. Correlación de Pearson – Punto 2 (Tanque 1) 26
5.3.3. Correlación de Pearson – Punto 3 (Tanque 5) 27
5.4. Análisis de los resultados usando Coeficiente de determinación R^2 . . 28
5.4.1. Coeficiente de determinación R^2 – Punto 1(Influente) 28
5.4.2. Coeficiente de determinación R^2 – Punto 2 (Tanque 1) 29
5.4.3. Coeficiente de determinación R^2 – Punto 3 (Tanque 5) 29
5.5. Análisis de los resultados usando la t-Student 29
5.6. Limpieza de Datos 30

6. Diseño de la Red Neuronal 31
6.1. Multi Layer Perceptron (MLP) 32
6.1.1. Red para datos obtenidos del sistema de control en lazo abierto 32
6.1.1.1. Diseño y optimización de la red MLP 32
6.1.1.2. Entrenamiento y configuración de la red MLP 32
6.1.2. Red para datos obtenidos del sistema de control en lazo cerrado 34
6.1.2.1. Diseño y optimización de la red MLP 34
6.1.2.2. Entrenamiento y configuración de la red MLP 35
6.2. Long-Short Term Memory (LSTM) 36
6.2.1. Red para datos obtenidos del sistema de control en lazo abierto 36
6.2.1.1. Diseño y optimización de la red LSTM 36
6.2.1.2. Entrenamiento y configuración de la red LSTM . . . 37
6.2.2. Red para datos obtenidos del sistema de control en lazo cerrado 39
6.2.2.1. Diseño y optimización de la red LSTM 39
6.2.2.2. Entrenamiento y configuración de la red LSTM . . . 39

7. Resultados y Discusión 41
7.1. Comparación de desempeño en lazo abierto 41

7.2. Comparación de desempeño en lazo cerrado 43

8. Conclusiones y Recomendaciones **45**

Referencias **47**

Índice de figuras

1. Elementos de una Neurona [23]. 7
2. Esquema de una Neurona Artificial [23]. 7
3. Estructura de una Red Neuronal Simple [23]. 8
4. Estructura de una red neuronal de una capa [23]. 11
5. Red neuronal de dos Capas[23]. 12
6. Procedimiento de entrenamiento de una ARN [17]. 13
7. Estructura estándar de una red LSTM [22]. 14
8. Función de activación ReLU. 15
9. Proceso de fangos activados [24]. 21
10. Visión general de la planta BSM1 [2]. 21
11. Coeficiente de correlación de Pearson – Punto 1(Influente). 26
12. Coeficiente de correlación de Pearson – Punto 2 (Tanque 1). 26
13. Coeficiente de correlación de Pearson – Punto 3 (Tanque 5). 27
14. Procesos biológicos reflejados en al ASM1 [1]. 27
15. Coeficiente de determinación R^2 – Punto 1(Influente). 28
16. Coeficiente de determinación R^2 – Punto 2 (Tanque 1). 29
17. Coeficiente de determinación R^2 – Punto 3 (Tanque 5). 29
18. Estructura de la red neuronal MLP. 32
19. Progreso de entrenamiento MLP. 33
20. Estructura de la red neuronal MLP. 34
21. Progreso de entrenamiento MLP. 36
22. Estructura de la red neuronal LSTM. 37
23. Progreso de entrenamiento LSTM. 38
24. Estructura de la red neuronal LSTM. 39
25. Progreso de entrenamiento LSTM. 40
26. Resultados de comparación de predicciones vs resultados reales. . . . 42
27. Resultados de comparación de predicciones vs resultados reales. . . . 42
28. Resultados de comparación de predicciones vs resultados reales. . . . 44
29. Resultados de comparación de predicciones vs resultados reales. . . . 45

Indice de tablas

1. Variables tomadas de la simulación de la planta 24
2. Detalle de cálculo para coeficiente de correlación de Pearson 25
3. Detalle de cálculo para coeficiente de determinación R^2. 28
4. Resultados prueba t-Student en lazo cerrado para el punto 1 (Influente). 30
5. Resultados prueba t-Student en lazo cerrado para el punto 2 (tanque 1). 30
6. Resultados prueba t-Student en lazo cerrado para el punto 3 (Tanque 5). 30
7. Hiperparámetros MLP para lazo abierto. 33
8. Hiperparámetros MLP para lazo cerrado. 35
9. Hiperparámetros LSTM para lazo abierto. 37
10. Hiperparámetros LSTM para lazo cerrado. 40
11. Hiperparámetros de las redes MLP y LSTM en modo lazo abierto . . 41
12. Comparación de los hiperparámetros de los modelos de redes neuronales para lazo cerrado. 43

1. Introducción

El crecimiento de la población y el aumento de industrias globales en la actualidad han sido los principales agravantes de la contaminación ambiental por las aguas residuales que generan sus actividades. Esto se ha convertido en una amenaza muy grave que, acompañada del cambio climático, genera aumentos de temperatura, fenómenos meteorológicos extremos, migración por parte del hábitat de la fauna, aumento de los mares, entre otros [9]. Las aguas residuales no solamente afectan al medio ambiente, sino también a la salud de los seres humanos, provocando enfermedades de interés público como la fiebre tifoidea o el cólera [9]. Por otro lado, la contaminación del agua podría conllevar que se depositen determinados nutrientes en los cuerpos acuáticos, promoviendo el crecimiento de ciertos organismos como las algas. Estos cuerpos de agua representan una amenaza a la fauna porque se crean zonas deficientes de oxígeno [9].

Frente a esta situación, tanto las empresas públicas como privadas como la sociedad deben emprender acciones conscientes para mitigar la contaminación del agua, lo que causaría un problema ambiental. Una de estas acciones involucraría la verificación de las concentraciones de contaminantes; conocido como tratamiento de aguas residuales (EDAR), dentro de las aguas residuales para preservar el entorno y los recursos naturales donde se riegan las aguas depuradas. En el EDAR se destinan estaciones depuradoras, para obtener un efluente de aguas residuales en el cual el ciclo de devolución del agua no tenga mayor afectación hacia el medio ambiente [9]. Sin embargo, este proceso requiere de la reducción máxima en costos operativos y la producción de lodos para ser viable económicamente [9]..

El proceso de EDAR se estructura en varias etapas y procesos de tratamiento que se vinculan entre sí. Se empieza por la etapa del pretratamiento, que consiste en la eliminación de residuos de tamaño grande del agua residual bruta. Se continúa con la etapa de tratamiento primario, donde se eliminan los sólidos considerados de menor tamaño, como son los sólidos sedimentables y la materia que flota en el efluente. Dentro de la etapa del tratamiento secundario, se efectúan ciertos procesos biológicos que eliminan sólidos refractarios, es decir, el lodo que no pudo ser eliminado dentro de la tapa anterior [9]. Es preciso mencionar que dentro del tratamiento secundario es recomendable llevar a cabo un procesamiento de separación en la digestión de los lodos, reduciendo el volumen de residuos sólidos y estabilizando los materiales orgánicos. Esto último posibilita una mejor manipulación dentro del proceso y, por ende, se reducen los costos de almacenamiento y bombeo. Generalmente, se utiliza la degradación microbiana como proceso habitual para esta finalidad. Cada tratamiento se clasifica como tratamiento anóxico, aeróbico o anaeróbico [9]. Finalizadas estas etapas, se puede liberar el agua al medio ambiente ya que, como última etapa de tratamiento terciario, se encuentra dentro del efluente secundario, donde se elimina el nitrógeno y el fósforo, dando cumplimiento a la norma requerida [9].

El correcto control de los sistemas de agua se ha convertido en un reto, debido a la presencia de una gran cantidad de variables interrelacionadas y su alto grado de no linealidad. Además, la falta de datos disponibles y que estos sean precisos han

dificultado la creación de controladores efectivos de aguas residuales. Por ello, la creación de controladores inteligentes, basados en aprendizaje profundo, ha llegado a ser una de las promesas que puede optimizar el control de los sistemas acuáticos. El surgimiento de controladores inteligentes forma parte de la ola de inteligencia artificial, cuyo objetivo principal es imitar varias de las funciones cognitivas de los seres humanos a través de modelos computacionales. Dentro de los procesos de control la inteligencia artificial ha aportado con varias técnicas y métodos que hoy en día son de gran utilidad para esta finalidad, donde las denominadas redes neuronales artificiales (RNA) constituyen uno de los métodos propuestos por la inteligencia artificial.

Las RNA han generado un gran potencial para la realización de modelos matemáticos precisos de procedimientos que son de mayor complejidad mejorando así el rendimiento de sus funciones [9]. En esta investigación, se consideran RNA para modelar el comportamiento de un sistema de aguas residuales. El modelo propuesto considera los datos del influente y el efluente de una planta [9].

Dentro de los últimos años, los horizontes se han ampliado en torno a la aplicación del aprendizaje profundo para mejorar el control de aguas residuales. Particularmente, las redes neuronales artificiales profundas han demostrado ser herramientas con capacidad de modelar sistemas de mayor complejidad y que no son lineales, basados en la utilización de datos antiguos y actuales. Estas herramientas permiten obtener patrones en los que existen relaciones no lineales en los datos, facilitando la realización de predicciones para el control de procesos complejos dentro del tratamiento de aguas residuales.

Por ello, la finalidad de esta investigación de master es diseñar y poner a prueba las arquitecturas de redes perceptrón multicapa y larga memoria de corto plazo.

1.1. Objetivos

1.1.1. Objetivo general

Desarrollar y aplicar una metodología de benchmarking para evaluar la eficacia de las redenes neuronales perceptrón multi-capa (MLP) y largo corto plazo (LSTM), basados en métodos de aprendizaje profundo, en el contexto del tratamiento de aguas residuales.

1.1.2. Objetivos específicos

Diseñar y poner a prueba arquitecturas de redes neuronales, incluyendo redes neuronales perceptrón multi-capa (MLP), redes neuronales de largo corto plazo (LSTM), en términos de su capacidad para modelar y predecir el comportamiento de los procesos de tratamiento de aguas residuales.

Analizar en profundidad el rendimiento de los predictores inteligentes, en diferentes condiciones climáticas (seco, lluvia, tormenta), con el fin de comprender a

fondo las capacidades y limitaciones de estos controladores en el control de sistemas de aguas residuales.

Mejorar la capacidad predictiva de los controladores inteligentes en el tratamiento de aguas residuales, con el objetivo de anticipar el comportamiento del sistema con una antelación de al menos 4 horas, y explorar cómo estas capacidades predictivas pueden ser utilizadas para mejorar la eficacia del control en tiempo real.

2. Estado del arte

En la presente investigación se realiza una revisión bibliográfica para recopilar y analizar los diferentes aportes teóricos existentes sobre redes neuronales artificiales profundas para el tratamiento de aguas residuales. Para este fin, se revisaron diversos repositorios académicos, que se describen a continuación.

2.1. Aplicación de controles inteligentes distribuidos

En la literatura científica, se han llevado a cabo numerosos estudios de modelación en el contexto del tratamiento de aguas residuales, con el objetivo de optimizar la eficiencia, la calidad y la sostenibilidad de dichos procesos. Estos estudios han empleado diversos enfoques de modelación, como modelos matemáticos de regresión multivariable, la metodología de superficie de respuesta, el uso de software especializado y redes neuronales artificiales. Estas herramientas de modelación se han utilizado con diversos propósitos, como comprender mejor los procesos de tratamiento, evaluar el impacto de variables en la calidad del tratamiento y analizar el efecto de las dosificaciones de coagulante y floculante en la turbidez del agua tratada [8]. Estos estudios han contribuido significativamente al avance del conocimiento en el campo del tratamiento de aguas residuales y han proporcionado información valiosa para la mejora continua de los procesos de tratamiento en busca de un mejor rendimiento y una mayor eficiencia en la eliminación de contaminantes.

Además de los enfoques de modelación mencionados anteriormente, también se han explorado otras estrategias para mejorar el tratamiento de aguas residuales mediante el uso de controladores distribuidos inteligentes como una estrategia prometedora para mejorar la eficiencia y la calidad de los procesos de tratamiento de agua [6], [19] . Por ejemplo, se han empleado algoritmos de control adaptativo y técnicas de optimización para ajustar dinámicamente los parámetros de los procesos de tratamiento en función de las condiciones cambiantes [14]. Como lo menciona Therrien et al. (2020) [31] estos enfoques permiten una respuesta más rápida y precisa a las variaciones en la carga de contaminantes y en las características del agua residual, lo que a su vez mejora la eficiencia y la efectividad del tratamiento. Además, se han investigado sistemas de control basados en la lógica difusa y la inteligencia artificial, los cuales pueden adaptarse y aprender de manera autónoma para optimizar continuamente el rendimiento del tratamiento [26], [20]. Estas estrategias innovadoras están demostrando ser prometedoras para el tratamiento de aguas residuales y

están abriendo nuevas oportunidades para alcanzar niveles más altos de eficiencia y sostenibilidad en este campo.

Los controladores distribuidos inteligentes se utilizan para monitorear y ajustar en tiempo real los parámetros de calidad del agua, como el pH, la concentración de oxígeno disuelto y la turbidez. Esto permite mantener los niveles de calidad del agua dentro de los límites establecidos y garantizar el cumplimiento de las normativas ambientales [6]. El potencial de los controladores distribuidos inteligentes en el contexto del tratamiento de aguas residuales ofrece una mayor flexibilidad, eficiencia y capacidad de adaptación en comparación con los enfoques de control tradicionales [6]. Sin embargo, aún existen desafíos técnicos y prácticos que deben abordarse, como la selección adecuada de los algoritmos de aprendizaje automático, la integración con los sistemas existentes y la aceptación y adopción por parte de los operadores y gestores del tratamiento de aguas residuales.

2.2. Aprendizaje profundo

El método de Aprendizaje Profundo (AP) a nivel global ha sido uno de los fundadores para la automatización directa de procesos tomando en cuenta a las redes neuronales multicapa como una de sus principales creaciones, lo que permite en la actualidad poder realizar varias acciones, como el reconocimiento de voz, el reconocimiento visual y una gran cantidad de dominios [13]. Por otro lado, el AP se ha encargado de utilizar algoritmos de retro propagación, determinando los cambios de las funciones de los equipos internos manejados mediante la sistematización de redes neuronales capa por capa [12]. Dentro del aprendizaje profundo se encuentran diversas creaciones de redes neuronales artificiales (RNA), lo que ha permitido establecer un gran avance a nivel global.

El uso del aprendizaje profundo en el control de procesos, específicamente en las plantas de tratamiento de aguas residuales, ha ganado popularidad en los últimos años. Países desarrollados han investigado y aplicado diversas tecnologías, incluyendo sistemas de control automático y modelos basados en inteligencia artificial, para optimizar la eficiencia de estas plantas [15]. Se han desarrollado y validado modelos de inteligencia artificial, como las redes neuronales, para mejorar la operación y gestión de las plantas de tratamiento de aguas residuales. Estos modelos han demostrado ser eficaces en la predicción de indicadores clave como la Demanda Química de Oxígeno (COD) en el efluente de los reactores de tratamiento [15]. La aplicación de redes neuronales ha sido utilizada para predecir el valor de COD del efluente en un reactor de inteligencia computacional (IC), que es una forma específica de tratamiento anaeróbico de aguas residuales. Los factores de entrada para el modelo incluyen la temperatura del agua, el pH del influente, el flujo del influente, el COD del influente y los Ácidos Grasos Volátiles (VFA) del influente. El modelo de red neuronal se entrena utilizando los datos operativos de una planta de tratamiento de aguas residuales. Una vez entrenado, el modelo se valida y se optimiza antes de ser probado con un conjunto de datos de prueba.

2.3. Predicción de la calidad del agua con redes neuronales MLP y LSTM

El modelo de memoria de largo y corto plazo conocido como (LSTM), está compuesto por un grupo de neuronas que son periódicas, implementadas para solución de la desaparición o daño de las gradientes lo cual ha comprometido el correcto funcionamiento de las redes neuronales periódicas, este modelo en conjunto con el Perceptrón Multicapa (MLP), que está basado en el algoritmo de retropropagación, es decir, este modelo puede retroceder para realizar cambios de pesos o sesgos en una red neuronal artificial. Dichos modelos han tomado el control para mejorar características internas en varios campos de los que se quieran investigar.

Para el caso del control y la gestión efectiva de la calidad del agua en los sistemas de alcantarillado, es fundamental realizar un abordaje sobre los desafíos del tratamiento de aguas residuales en entornos urbanos. Es por ello que se ha llevado a cabo una investigación para explorar el uso de redes neuronales profundas, (MLP) y (LSTM), en la predicción de la calidad del agua en sistemas de alcantarillado. Estas técnicas de aprendizaje automático han demostrado su capacidad para capturar patrones complejos y no lineales, lo que permite una mayor precisión en la predicción y el control de la contaminación [11].

Los estudios revisados revelan que las redes MLP han sido ampliamente aplicadas en la predicción, demostrando su capacidad para modelar relaciones complejas entre múltiples variables, como pH, conductividad, BOD_5, COD, $NH_4^+ - N$, TN, TP, entre otros [16]. Al entrenar una MLP con datos históricos, se pueden obtener predicciones precisas sobre los cambios futuros. Sin embargo, se han identificado desafíos, como el sobreajuste y la optimización de los hiper parámetros, que requieren atención para mejorar el rendimiento y la generalización de los modelos de MLP en este contexto [11].

Por otro lado, las redes LSTM han sido capaces de capturar y modelar patrones secuenciales y dependencias a largo plazo en los datos obtenidos. Al considerar la naturaleza dinámica de los sistemas de alcantarillado y la variabilidad temporal de los indicadores de calidad del agua, las LSTM han demostrado ser efectivas en la predicción en diferentes escalas temporales [3]. Por lo tanto, se requiere una cuidadosa selección y ajuste de los parámetros de las LSTM para evitar problemas de sobreajuste y garantizar un rendimiento óptimo [11]. El uso de redes neuronales profundas, como las MLP y las LSTM, en la predicción de la calidad del agua en sistemas de alcantarillado ha mostrado resultados prometedores en términos de mejora de la precisión y el control de la contaminación. Estos enfoques basados en datos han ganado popularidad debido a su capacidad para capturar patrones complejos y no lineales en [3]. Para la implementación exitosa de estos modelos requiere considerar desafíos específicos, como la selección adecuada de arquitecturas de red, la optimización de parámetros y la disponibilidad de datos en tiempo real. Además, es necesario abordar desafíos adicionales, como la integración de sistemas de monitoreo en tiempo real y la aplicación de estrategias de regularización para mejorar la generalización de los modelos [11].

2.4. Optimización del simulador BSM1 mediante control predictivo

El Benchmark de Simulación (BSM1) ha revolucionado el campo de la investigación en el tratamiento de aguas residuales al proporcionar un marco común y estandarizado para la evaluación comparativa de diferentes estrategias de control [10]. Una de las técnicas más destacadas que se ha aplicado al BSM1 es el Control Predictivo basado en Modelos (MPC) [7]. El BSM1 proporciona un modelo estandarizado y de gran escala de una planta de tratamiento de aguas residuales, incluyendo un reactor biológico, un tanque de sedimentación secundaria, y sistemas de reciclado de lodos [10]. Esta herramienta de simulación permite a los investigadores y profesionales analizar el rendimiento de distintas estrategias de control bajo condiciones operativas idénticas, facilitando así la comparación objetiva de diferentes enfoques.

El modelo BSM1, ha sido utilizado para controlar varios aspectos del proceso de tratamiento de aguas residuales [7]. Esto se logra a través de la predicción y optimización del comportamiento futuro del sistema basándose en un modelo de predicción, que suele ser un modelo de espacio de estados lineal y discreto del sistema [18].Tiene la capacidad para manejar variables y restricciones que han demostrado ser particularmente útiles en el entorno complejo de una planta de tratamiento de aguas residuales [21]. Además, la sintonización del controlador es un aspecto crítico en la implementación del MPC en el BSM1 [7]. La determinación de pesos para el cambio de las variables de control y los errores de seguimiento del punto de ajuste es un desafío considerable, y se han desarrollado diversos métodos para la sintonización automática del MPC.

A pesar de los desafíos, la combinación del BSM1 y el MPC ha demostrado ser una valiosa dirección de investigación en el campo del control de procesos de tratamiento de aguas residuales [7]. Los esfuerzos futuros probablemente se centrarán en mejorar la sintonización y el rendimiento del MPC en el marco del BSM1, así como en explorar nuevas técnicas y enfoques para el control de este tipo de sistemas.

3. Redes neuronales

En la actualidad las redes neuronales se han convertido en un gran aporte para la inteligencia artificial. Además de ello, muchos investigadores mencionan que son redes que se encuentran preparadas para las entradas que se obtienen de escenarios internos o externos dentro del sistema y estas están asignadas por pesos al azar [23]. Se puede afirmar que las redes neuronales representan un conjunto de técnicas de procesamiento de la información del sistema nervioso biológico de un ser vivo, lo que se intenta es emular el cerebro del ser humano, siendo capaz de resolver funciones no lineales en corto tiempo ya que aprenden de información que es difícil de expresar matemáticamente y por esta razón las RNA se han convertido en herramientas poderosas para la modelación de sistemas y análisis de señales.

3.1. Redes neuronales artificiales

El elemento procesador es la unidad similar a la neurona del ser humano dentro de las redes neuronales artificiales [5], las entradas de un elemento procesador se pueden combinar mediante una suma básica, dicha suma puede ser modificada por una función de traslado, este valor resultante se dirije hacia la salida del elemento procesador [5].

La neurona es considerada como la unidad fundamental para el sistema nervioso y el correcto funcionamiento del cerebro. La neurona es un procesador que recibe y realiza una combinación de actividades desde y hacia otras neuronas [23]. Por lo tanto la entrada debe ser fuerte dentro de su combinación, para que la salida de la misma se active. A continuación, en la figura 1, mostraremos las partes que constituyen a una neurona.

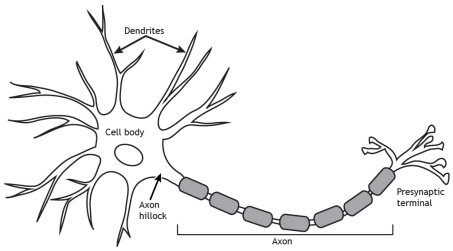

Figura 1: Elementos de una Neurona [23].

El cerebro humano se compone por billones de neuronas que se encuentran conectadas. Las Dendritas son las entradas de otras neuronas que tienen uniones llamadas sinapsis, las cuales están ramificadas hacia el axón, es decir, la salida de la neurona. [5]. Es importante saber, la eficacia que debe tener la sinapsis al ser moldeable dentro de la duración del estudio de la red.

Las salidas del elemento procesador pueden interconectarse hacia las entradas de algunos tipos de neuronas artificiales, mediante las uniones ponderadas, que corresponden al funcionamiento eficaz de la sinapsis neuronal [5]. La Figura 2 demuestra como se ve un PE de una red neuronal artificial implementada desde un ordenador.

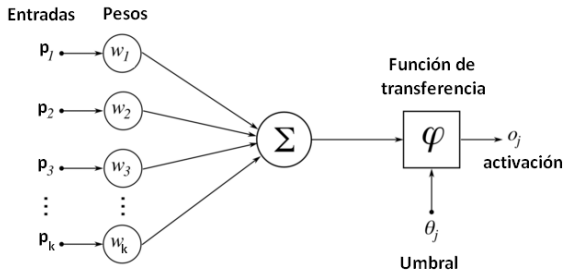

Figura 2: Esquema de una Neurona Artificial [23].

La red neuronal está compuesta por una serie de procesadores elementales que se encuentran conectados de manera concreta. Además del elemento procesador, la

forma en cómo se conectan las RNA tiene gran relevancia [32]. Los PE están estructurados en conjunto, se los denomina niveles o capas. Una red neuronal tradicional contiene en un grupo de capas con uniones entre capas anexadas.

Hay un par niveles que están conectados al entorno externo. Un nivel de entrada, que funciona como un almacenamiento temporal para los datos que se introducen en la red, y otro nivel de salida, que guarda una respuesta de la red ante una entrada determinada. Los demás niveles son intermedios y son conocidos como capas ocultas [32]. La apariencia de una Red Neuronal Artificial se observa en la Figura 3.

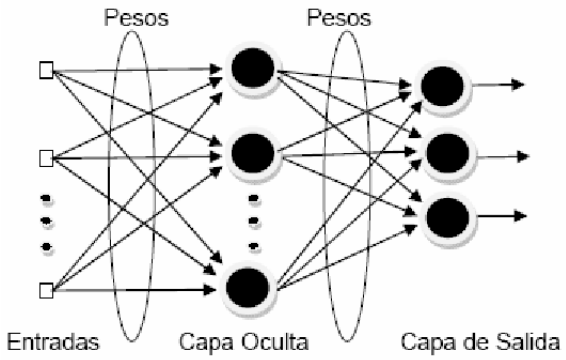

Figura 3: Estructura de una Red Neuronal Simple [23].

3.1.1. Características de las redes neuronales artificiales

El funcionamiento de las redes neuronales artificiales se ha basado en el comportamiento de las neuronas de un cerebro humano. Dichas neuronas se conforman por ciertas características que simulan a una neurona orgánica. Dentro de las tareas más frecuentes de una red neuronal artificial es la capacidad de adquirir conocimientos a partir de la experiencia, por lo que toman en cuenta varios de los factores principales de una serie de datos [23].

- **Aprender:** Las RNA han logrado alcanzar el conocimiento sobre cualquier tema, mediante una previa investigación. De la misma manera, aprenden de los datos. Para ellos se presenta una serie de entradas para que ellas mismas realicen los ajustes necesarios para establecer unas salidas consistentes.

- **Generalizar:** Las RNA generalizan su conocimiento, ya que es propiamente de su estructura y naturaleza. Dichas redes ofrecen resultados exactos a ciertos accesos o entradas, que presentan cambios pequeños a causa del impacto que produzca la distorsión o el ruido.

- **Aislar:** Existen algunas RNA que tienen la facultad de extraer patrones de los datos y aislarse en una serie de entradas que al parecer no contienen características en común.

3.1.2. Aplicaciones de las redes neuronales artificiales

Para las redes neuronales artificiales existen algunas propiedades únicas que permiten a los ordenadores de manejo neuronal establecer una nueva técnica de cálculo, de tal manera que se pueda emplear a ciertas aplicaciones. El manejo neuronal mediante ordenadores establece un enfoque que se acerca más a las reacciones y percepciones humanas en comparación con otros métodos convencionales. [32]. Las redes neuronales artificiales ocasionan como consecuencia situaciones aceptables, ya que las entradas se encuentran incompletas. Las áreas de aplicación de las redes neuronales artificiales se han dedicado al procesamiento de señales, estudio de imágenes, control de procesos, procesamiento del lenguaje integrando la robótica, etc. Por otro lado, las RNA, han aplicado a las siguientes investigaciones:

- **Transformación de Texto a Voz:** Terrence Sejnowski se convirtió en uno de los principales motivadores para el manejo neuronal mediante ordenadores en el campo de la trasnformación de texto a voz. Actualmente se lo denomina Nettalk,que se encarga de convertir las letras o símbolos en un lenguaje hablado, con la ayuda de un sintetizador de voz llamado Dectalk. La principal ventaja de este descubrimiento es que eliminar la necesidad implmentar campos de programación complejos integrando algunas reglas de pronunación en los ordenadores.[30]. Este descubrimiento ha sido motivo de seguimiento en el avance de la investigación para el desarrollo comercial.

- **Procesamiento Natural del Lenguaje:** Los científicos McClelland y Rumelhart son los principales desarrolladores de una red neuronal artificial basada en el procesamiento del lenguaje verbal, en este caso se ha enfocado en el idioma inglés, utilizaron en un principio la generación de pronósticos de verbos desconocidos o nuevos [32]. Para la identificación de las gráficas, se utilizan datos que ocupan menos memoria, los investigadores Cottrel, Munro y Zisper crearon una RNA que se enfoca en la compresión 8:1 para el reconocimiento de todo tipo de imágenes.[30].

- **Identificación de Símbolos:** Nestor, Inc fue uno de los investigadores encargados en el reconocimiento de caracteres mediante la creación de una RNA que esta enfocada en identificar y clasificar todo tipo de símbolos de manera exitosa.

- **Identificación de modelos en Gráficas:** Las aplicaciones habituales de identificación de objetivos detectados ha desarrollado varias redes neuronales artificiales que utilizan el método de Backpropagation, el cual ha demostrado un comportamiento similar al de los operadores humanos en términos de clasificación.

- **Desarrollo de soluciones para problemas de combinatoria:** El cálculo convencional para encontrar una solución a este tipo de problemas necesita de un tiempo de procesamiento que aumenta de forma acelerada con la cantidad de entradas. [30]. Hopfield desarrolló con con buenos resultados, una red

neuronal artificial que abarca soluciones a este tipo de problemas, como por ejemplo mejorar las rutas a los viajeros y establecer los caminos más cortos.

- **Procesamiento de señales abordadas con RNA:** Lapedes y Farber son los científicos que han demostrado que la red neuronal backpropagation puede realizar una correcta,predicción, modelado de sistemas y filtrado de ruido. En el caso de la predicción, muchas veces se cuenta con una serie temporal de datos o valores que nos permiten conocer el comportamiento de un fenómeno en el mundo real.[30].

- **Modelación de Sistemas:** Las redes lineales se definen gracias su función de transmisión, que combina un factor de salida con un factor independiente y sus resultantes, mediante una expresión matemática. Por su parte, las RNA tienen la capacidad de estudiar una función de transmisión y tener el comportamiento adecuado en la modelización de sistemas lineales.

- **Filtro de Ruido:** Los sistemas de redes neuronales artificiales también pueden enfocarse para filtrar las señales, logrando preservar en gran medida las estructuras y valores que se obtienen con los depuradores tradicionales.

- **ServoControl:** Se enfrenta un problema complejo en el control de un sistema de servomecanismos, que se basa en la búsqueda una regla de cálculo computarizado viable para equilibrar los cambios físicos que suceden dentro del sistema. Uno de los principales problemas, es la falta de precisión para medir las modificaciones y el tiempo de cálculo excesivo necesario para la obtención de una solución matemática [32]. Las RNA han sido preparadas para predecir el error obtenido en la presentación final de la posición del robot, lo cual se va a combinar con una serie de correcciones que se adapten a la localización final, mejorando así su precisión,

3.1.3. Redes neuronales profundas

Las redes neuronales de una capa oculta se ven obligadas a realizar funciones continuas de alta variabilidad, que, al momento de modelarlas, necesitan una cantidad exuberante de neuronas, generando así la necesidad de implementar arquitecturas profundas, las cuales contienen gran cantidad de capas con componentes adaptativos no lineales, permitiendo representar una variedad de funciones de forma conjunta. Existe un método convencional básico que consiste en aplicar valores aleatorios para ser ajustados por los pesos de las redes neuronales, dicho método se lo considera pobre por la existencia de varios obstáculos. Basándose en los problemas existentes, en el año 2007, se propone la etapa de pre entrenamiento de cada capa en forma sistemática, que consiste en un entrenamiento no supervisado mediante máquinas de Boltzmann restringidas, permitiendo dar inicio a los parámetros establecidos para las redes neuronales profundas.

3.2. Red perceptrón multicapa (MLP)

La habilidad de procesamiento y rendimiento de las redes neuronales artificiales se debe a las interconexiones entre las neuronas artificiales que forman la estructura de las RNA. La configuración más elemental consiste en un conjunto de neuronas ordenadas por una única capa, la cual se puede apreciar en la figura 4, [32]. Las uniones circulares no participan como componentes de la capa, sino que simplemente distribuyen las entradas recibidas. Mientras que una red formada de múltiples capas, como se observa en la figura 5 tendrá la capacidad de resolver problemas que no son linealmente separables, ya que esa sería su principal limitación.

3.2.1. Estructura de la red perceptrón multicapa

Las neuronas artificiales se conectan a la red neuronal por medio de un peso específico o entrada. Dentro de la práctica, existen uniones a través de las salidas o de las entradas de las RNA en una sola capa. Normalmente, es claro que las redes con gran tamaño y complejidad generan un mayor rendimiento para establecer un cálculo computarizado en comparación con las redes básicas.[30].

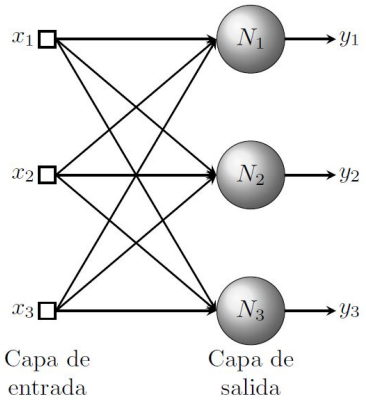

Figura 4: Estructura de una red neuronal de una capa [23].

A pesar de los cambios existentes en las redes construidas, pueden presentar detalles con ciertas diferencias, por otro lado, tienen similitud en la distribución de las neuronas en capas, como se aprecia en la figura 5, reproduciendo la estructura de capas localizadas en el cerebro humano [30]. Las redes neuronales artificiales multicapa están constituidas por algunas capas sencillas interconectadas en forma de cascada, las cuales tienen ventajas y características superiores a las RNA de una sola capa.

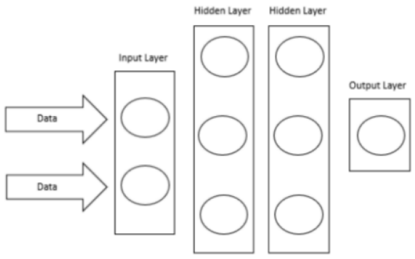

Figura 5: Red neuronal de dos Capas[23].

Es importante resaltar que la presencia de una labor de estimulación no lineal entre varias capas, mejora con mayor efectividad el funcionamiento de las RNA multicapa.

3.2.2. Proceso de aprendizaje de la red perceptrón multicapa

Las redes neuronales artificiales están compuestas por un conjunto de neuronas que tienen conexión entre ellas, la relación de las conexiones neuronales son valoradas basándose en su peso, lo que se relaciona por el múltiplo del valor de entrada. Dentro del proceso general se empieza por la activación neuronal, como la función sigmoidal, tangente hiperbólica, entre otras, para determinar su salida. La función de activación es esencial, ya que introduce la no linealidad durante la preparación de la red, lo cual es importante para el modelado efectivo de las redes neuronales artificiales.

Durante la preparación de una red neuronal, como en la figura 6, se lleva a cabo un ciclo iterativo en el que se traspasan capa por capa de cada RNA. Durante este ciclo, satisfaciendo de esta manera las necesidades del modelo mediante el ajuste de los valores basado en los pesos y sesgos. El proceso de propagación hacia adelante, también conocido como forward-propagation, ocurre cuando la información de la entrada es transferida por medio de la red.

En este proceso, cada neurona se adapta y aprende de los datos obtenidos por cada capa anterior, luego envía esta información a la siguiente capa. Después de que toda la información haya pasado por todas capas y a su vez las neuronas siempre y cuando hayan realizado sus cálculos, llegando así a la última capa, donde se puede determinar cada entrada mediante un pronóstico de etiqueta.

Las pérdidas son utilizadas para calcular el error, por lo tanto, se puede realizar una comparación cuantitativa verificando un resultado correcto, buscando como valor final igual a 0, luego, mediante una preparación del modelo, se realiza un ajuste automático necesario para las RNA y por último obtener el mejor pronóstico posible.

Una vez que se ha calculado el error en la capa de salida de la red neuronal, este error se propaga hacia atrás, retro-propagándose (back-propagation), a lo largo de la red. Dicho error es transportado desde la salida hacia las neuronas que contienen capa oculta y están interconectadas con la salida. Las capas que se encuentran ocultas obtienen una parte del valor erróneo, tomando en cuenta su aportación a la

salida final. El proceso se reitera en varias ocasiones de capa en capa, asegurándose de que las neuronas reciban el fragmento de error equivalente a su participación del error total. Es decir, las capas ocultas aprenden a través de la retro-propagación del error para ajustar sus pesos y mejorar la precisión y capacidad de predicción de la red neuronal.

Luego que el error se ha propagado a través de la red neuronal, es el momento de adaptar el volumen de los valores resultantes de las conexiones con las neuronas. Para obtener un error aceptable debe ser cercano a 0, por lo que se aplica un modelo matemático conocido como decadencia por gradiente. Este modelo matemático realiza pequeños ajustes en los pesos mediante un cálculo de la derivada en función de la pérdida. Esto va a indicar la dirección en la cual se debe avanzar para descender hacia el mínimo global de la función de pérdida. El procedimiento de la decadencia por gradiente se repite en varias ocasiones, conocidas como etapas, y se realiza utilizando bases de datos. Esto implica que se toma un conjunto de datos, se calcula el error, se ajustan los pesos y se realiza nuevamente el proceso hasta que se alcanza un nivel de convergencia satisfactorio.

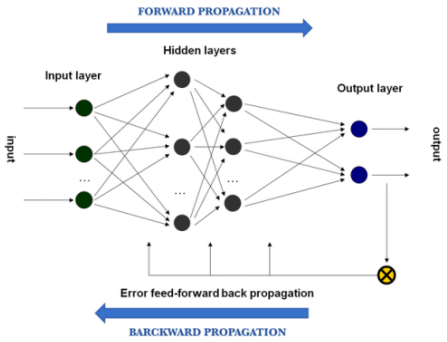

Figura 6: Procedimiento de entrenamiento de una ARN [17].

3.3. Long-short term memory (LSTM)

El modelo de memoria utilizado para corto y largo plazo está basado ampliamente en las redes neuronales recurrentes, que, por lo general, toman sus recuerdos para aprender de experiencias importantes que han sucedido hace mucho tiempo. Las LSTM permiten a la red neuronal reconocer sus entradas.

3.3.1. Estructura de la red LSTM

El modelo LSTM utiliza una estructura de memoria en lugar de unidades neuronales simples. Se compone de una celda de memoria, una puerta de entrada y una puerta de salida. Las puertas controlan el flujo de información, protegiendo a la celda de memoria de perturbaciones por datos de entrada irrelevantes y evitando que las perturbaciones se propaguen a otras neuronas.

En la figura 7 se presenta la configuración de una LSTM, en la cual los símbolos "x" y "+" representan multiplicaciones y sumas de elementos, respectivamente. Los bloques están representados por capas, mientras que "tanh" indica una función de tangente abultada. Además, "x" representa la entrada, "h" denota la salida y el estado en el que se encuentra [22].

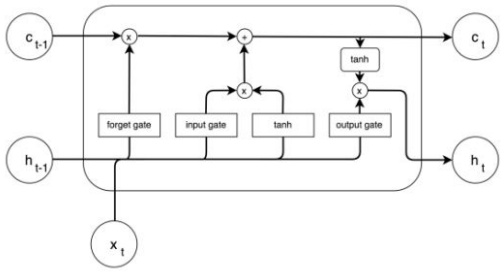

Figura 7: Estructura estándar de una red LSTM [22].

El diseño de las redes LSTM pueden recordar eventos a lo largo de períodos de tiempo arbitrarios, superando así las limitaciones de las redes neuronales recurrentes convencionales [22].

3.3.2. Proceso de aprendizaje de la red LSTM

Las aplicaciones que tienen las redes LSTM se utilizan comúnmente para el manejo de robots, por su capacidad de predicción del tiempo, reconocimiento de voz, etc [22]. Otra de las cualidades de este tipo de redes, es la habilidad de recordar valores en tiempos prolongados, lo que determina varios atributos de secuencia de entrada.

Para la función de activación del estado del modelo LSTM, su derivada será 1, por lo que al momento de la repropagación no tendrá daño alguno, y su salida tiene que ser en un rango de 0 a 1.

La ecuación de una red LSTM estándar es la que se presenta a continuación:

$$y = s\Big(\sum_{i=1}^{N} w_i x_i\Big) \tag{1}$$

En dicha ecuación s, representa la activación de la logística de la red.

El algoritmo de retropropagación a través del tiempo (BPPT) es el adecuado para el entrenamiento del modelo LSTM. La fuga de los gradientes está basada por el volumen de errores que se produzcan. El modelo recuerda sus errores mediante su función, su propagación hacia atrás, reparando así cada error en cada peso, por lo que considera que la repropagación es muy efectiva para acondicionar de forma correcta una red neuronal artificial.

3.4. Funciones de activación neuronal

Las funciones de activación neuronal desempeñan un papel fundamental al propagar hacia adelante la salida de una neurona hacia las neuronas de la capa siguiente a las que está conectada. Estas funciones son esenciales para no implementar la continuidad en el proceso de preparación del modelo de las redes neuronales artificiales, permitiendo que la red aprenda y represente relaciones complejas entre los datos. Existen varias funciones que activan las redes neuronales que se describen a continuación.

3.4.1. Activación unidad lineal rectificada (ReLU)

La tarea de activación de unidad rectificada uniforme (RELU) es considerada como una herramienta de aprendizaje profundo encargada de la activación no lineal, Krizhervsky logró popularizar la función ReLU, ya que ha sido una de las funciones con mayor utilización debido a su eficacia y simplicidad [33].

La función unidad rectificada uniforme se basa en la siguiente ecuación: $f(x) = max(0, X)$, donde x representa la neurona y la salida se representa con base en la cantidad máxima que existe entre 0 y x, es por ello que ReLU se basa en cualquier entrada x, sin importar si x es positivo o negativo siempre se devolverá 0 como se aprecia en la figura 8.

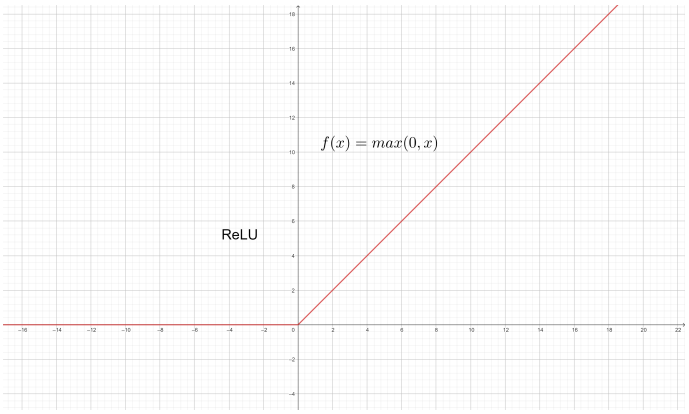

Figura 8: Función de activación ReLU.

Las ventajas que la convierten en una herramienta esencial dentro del aprendizaje profundo de las redes neuronales es, la facilidad de cálculo y el no necesitar de mucho poder para su procesamiento, tiene menos probabilidad con el problema de desvanecimiento de la gradiente, que es producida cuando la derivación de la función de activación es muy pequeña, dificultando la actualización los pesos ReLU, finalmente es utilizada dentro de las redes neuronales profundas, ya que no tiene saturación dentro de la activación lo que ocurre con otras funciones.

3.4.2. Activación sigmoide

La tarea de activación sigmoidal es útil para reducir la influencia de valores muy altos o diferentes, sin suprimirlos por completo. Al aplicar una función sigmoidal, las variables independientes que tienen un rango casi infinito se convierten en probabilidades simples que van desde 0 hasta 1. Con esta función, los valores tienden a acercarse en su mayoría a la salida, por lo tanto, estará muy cerca de los extremos de 0 o 1. La activación sigmoidal produce valores de salida que se encuentran en términos ya sean altos o bajos dentro del sigmoide. Como resultado, cuando la inclinación de la función es alta, es similar a una función escalón. No obstante, la activación sigmoidal es valiosa debido a que su derivación tiene resultado positivo y se acerca a cero para cantidades grandes, ya sean positivas o negativas, o cuando su valor más alto resulta ser igual a 0. Esto facilita utilizar parámetros de aprendizaje diseñados para funciones de gradiente, pero con la ventaja de la derivación que está basada en la función sigmoidal, la cual se encuentra establecida en todo el intervalo. Esta característica facilita la optimización de aprendizaje de la red neuronal, ya que la derivación es necesaria para calcular los gradientes y ajustar los pesos de manera adecuada.

3.4.3. Activación de tangente hiperbólica

La activación tangente hiperbólica (tanh), es utilizada cuando los datos a clasificar presentan cambios leves entre cantidades tanto positivos como negativos. Es una función muy comúnmente utilizada en preparaciones que se encuentran supervisadas, como ocurre en la retro-propagación del error. Sin embargo, es importante tener cuidado al utilizar esta función cerca de los límites positivos y negativos, ya que puede provocar que la salida siempre sea igual a 1 o -1, lo que se conoce como saturación.

3.5. Elementos del algoritmo de propagación hacia atrás

El backpropagation es un método que se utiliza para realizar cambios en cuanto al peso o al sesgo de una red neuronal en la dirección correspondiente. Empieza con el cálculo del valor de pérdida y luego encaja los factores de la red neuronal en orden contrario, utilizando una ecuación matemática de optimización que toma en cuenta el valor erróneo. En la herramienta de software, Matlab, existen tres ecuaciones de optimización diferentes disponibles:Declinación de gradiente aleatoria con momento (SGMD), RMSProp y Evaluación adaptativa del momento (Adam). , Para algunas investigaciones se utiliza el algoritmo de perfeccionamiento denominado, Adam, que es el más común para la preparación de redes neuronales.

3.5.1. Descenso de gradiente (GD)

Todos los métodos mencionados anteriormente se basan en el descenso por gradiente, que es uno de los algoritmos de mejora más comunes en aprendizaje automatizado y aprendizaje profundo. La declinación por gradiente innova los factores de la red neuronal, como los pesos y la dirección, con el propósito de disminuir la pérdida. Este algoritmo da reducidos movimientos en el camino contrario al gradiente de la pérdida, utilizando la primera derivación de la pérdida en cada repetición para determinar la inclinación de la función en un punto establecido.

$$\theta_{n+1} = \theta_n - \alpha \cdot \nabla E(\theta) \tag{2}$$

En la ecuación dada, n, simboliza el número de repeticiones, α es mayor a 0, el cual representa la agilidad de aprendizaje, θ es el vector de factores, y $E(\theta)$ es la pérdida. El gradiente de pérdida, $\nabla E(\theta)$, se calcula tomando en cuenta el grupo de preparación, y la ecuación de declinación de la gradiente tradicional, utilizando todos los datos del conjunto al mismo tiempo.

El procesodimiento implica realizar el cálculo de la derivación en base a la pérdida en cada capa oculta, utilizando la derivación de la pérdida de la capa superior y verificando la activación, capa por capa. Esto significa que todas las funciones de activación deben ser diferentes. Una vez que las neuronas en cada repetición, tienen el valor de la derivación de la pérdida respectiva, se restauran los factores en la dirección contraria al gradiente [17]. Para ser verdad, el gradiente apunta de forma continua en la dirección que aumenta la pérdida. Por lo tanto, al usar la gradiente en valores negativos, se puede alcanzar la dirección en la cual la pérdida se reduce.

Dentro del calculo del siguiente valor, la ecuación de declinación de la gradiente se adpata al valor real en dirección opuesta al gradiente, agregando un volumen directamente proporcional a este. Los valores de cambio se determinan por la cantidad del gradiente y un factor denominado tasa de aprendizaje. La ecuación para la declinación de la gradiente repite este procedimiento de forma repetitiva hasta llegar a un límite, en el cual la pérdida no pueda disminuir más [17].

3.5.2. Descenso de gradiente estocástico (SGD)

La ecuación del descenso de gradiente aleatoria tiene mayor rapidez y no necesita de muchos cálculos para renovar los factores de la red neuronal. En lugar de evaluar el gradiente en todo el conjunto de entrenamiento, se utiliza un subconjunto llamado minibatch. Cada minibatch se utiliza para calcular el gradiente y actualizar los parámetros en una repetición. Completar todo el proceso de preparación en todo el conjunto de actividades realizadas se utilizan minibatchs, denominados epoch. El proceso aprovecha operaciones eficientes en GPU para calcular simultáneamente el gradiente para varios ejemplos de entrada.

3.6. Parametrización de modelos

Los modelos existentes pueden mejorar los procesos de un sistema, por lo tanto para llevar a cabo un correcto funcionamiento, se requiere tener el conocimiento y tener las habilidades necesaria para la práctica, tomando en cuenta los ajustes de las variables adecuadas. Sin embargo, estas mejoras en la precisión pueden aumentar el tiempo necesario para el procedimiento de aprendizaje. Es por ello que encontrar la mejor configuración de parámetros y tareas de activación exigen habilidad y destrezas debido a la cantidad de medios disponibles.

Algunos de los parámetros más utilizados se describen a continuación.

3.6.1. Número de épocas

El parámetro "epochs" representa la cantidad de veces que todos los datos de entrenamiento son procesados por las redes neuronales a lo largo de su preparación.

3.6.2. Tamaño del lote

Como se mencionó anteriormente, es posible agrupar la información de preparación en bloques más pequeños, conocidos como mini-lotes (mini-batch), para alimentar la red neuronal. El tamaño del lote, denominado batch-size, determina la cantidad de muestras que se utilizarán en cada repetición de la preparación para renovar el gradiente. El tamaño adecuado del bloque depende de varios parámetros, incluyendo la técnica utilizada para realizar los cálculos, denominada memoria GPU.

3.6.3. Tasa de aprendizaje

El portador de gradiente tiene un sentido y un volumen. Las ecuaciones para la declinación de la gradiente es producto entre el tamaño de la gradiente por una escala, a esta ecuación se la denomina como tasa de aprendizaje (α).

$$\theta_{N+1} = \theta_n - \alpha \cdot \nabla E(\theta) \tag{3}$$

La cantidad adecudada para este factor tiene que ver con el modelo en cuestión. Generalmente, si el tamaño del lote es demasiado grande, se estarán dando movimientos de gran magnitud durante el proceso de aprendizaje. Aunque esto podría acelerar el entrenamiento, también existe el riesgo de no alcanzar el objetivo esperado de pérdidas, lo que dificultaría el procedimiento de aprendizaje podría afectar la necesidad del resultado esperado.

Por otro lado, si el learning rate es bajo, los avances serán constantes y pequeños, lo que permitirá una mejor aproximación al mínimo local. Sin embargo, esto también significa que el procedimiento de aprendizaje será más pausado. Normalmente, el learning rate óptimo es aquel que tiene un avance más lento cuando el modelo

comienza a acercarse a una solución, como es el caso de la aplicación del modelo en Matlab donde se puede utilizar el parámetro "LearnRateSchedule" con el valor "piecewise" para disminuir el learning rate a medida que avanzan los epochs [17].Esto facilita un proceso de enseñanza veloz en las etapas iniciales mediante la utilización de tasas de aprendizaje más elevadas, conforme se avanza en el entrenamiento, se aplican ajustes cada vez más sutiles, promoviendo la convergencia hacia el valor mínimo de la función de pérdida durante el período de entrenamiento.

3.6.4. Momento

Cuando se utiliza el algoritmo de descenso de gradiente estocástico, puede ocurrir cierta oscilación durante el proceso de caídc hacia el mínimo local óptimo. Sin embargo, se ha encontrado que al agregar un término de impulso a la actualización de los parámetros, esta oscilación puede reducirse. Se puede definir el descenso de gradiente estocástico con actualización de momentum de la siguiente manera:

$$\theta_{n+1} = \theta_n - \alpha \cdot \nabla E(\theta) + \gamma \cdot (\theta_n - \theta_{n-1}) \tag{4}$$

El momento, que es cómo la iteración actual se ve afectada por el paso del gradiente anterior, se representa con el símbolo ∇.

3.6.5. Recorte del gradiente

Cuando el gradiente aumenta rápidamente, el entrenamiento se vuelve inestable y diverge rápidamente. Esto se conoce como "explosión del gradiente" y se puede observar cuando la función de pérdida muestra valores "NaN" o "Inf". El gradiente indica la dirección de mayor crecimiento del error y ajustar los pesos en esa dirección puede reducir el error. Sin embargo, en redes neuronales con capas profundas, si los pesos adquieren valores altos al principio o durante el entrenamiento, el gradiente puede dispararse o debilitarse a medida que se propaga a través de la red debido a la influencia de todos los pesos [17]. En resumen, el gradiente se puede expresar como $\nabla E(\theta) = WT - 1$, donde W representa los pesos y T es el número de iteraciones del error. Por ejemplo, si W es igual a 1.5 y T es igual a 20, el gradiente tendría un valor de 3325.3, lo que se llama "explosión del gradiente" y se representa computacionalmente como "Inf" o "NaN". Por otro lado, si W es igual a 0.9, el gradiente tendría un valor de 0.1216, lo que se denomina "gradiente desvaneciente". Por lo tanto, es importante que los pesos estén cercanos a 1 para evitar estos efectos [17]. El parámetro de "recorte del gradiente" ayuda a prevenir la explosión del gradiente limitando su valor, lo que estabiliza el entrenamiento incluso con tasas de aprendizaje más altas y en presencia de valores atípicos.

4. Descripción y modelo matemático del proceso de fangos activados

4.1. Etapas de tratamiento de las aguas residuales

Dentro del trato adecuado de aguas residuales, tanto industriales como urbanas es necesaria la eliminación de todo tipo de desechos transportados. Es necesario aplicar ciertas operaciones físicas, químicas o biológicas de las cuales son agrupados en las que tienen que pasar por ciertas etapas tomando en cuenta un tratamiento inicial [17]. Dentro del tratamiento inicial se procede por un proceso de eliminación de todo tipo de sólidos que llegan del colector, siendo así como una última etapa se realiza una separación de grasas cumpliendo de esta manera dos metas que son importantes como es eliminar la contaminación y aplicar protección a la entrada de residuos inadecuados.

En la primera etapa tienen como objetivo principal eliminar las partículas que fueron retenidas dentro del tratamiento inicial, por lo que se utilizan ciertos químicos que realizan una agrupación de partículas. En la segunda etapa y considerada una de las más importantes se realiza la disminución de sustancia orgánica que se puede degradar de forma biológica, en donde contiene aguas residuales, tomando en cuenta que se quiere evitar cualquier tipo de producción de microorganismos en exceso. Dichos tratamientos contienen planta de tratamiento, que intenta reducir las características de los fangos, volumen y peso.

4.2. Descripción del Proceso de Fangos Activados

En la segunda etapa, se considera una de las más importantes se aplica el procedimiento de los fangos activados que tiene como objetivo la eliminación de bacterias existentes en los residuos orgánicos, suprimiendo también los microorganismos dentro del agua que se está tratando [29]. Siendo así que los fangos son considerados como un cultivo bacteriano aerobio. Por lo tanto, como se observa en la figura 9 se puede resumir que el tratamiento de fangos activados consiste en mezclarlos con un conjunto de microorganismos, lo cual generara lodos activados, para luego someterlo a aeración, lo que permite que los microorganismos descompongan la materia orgánica, reduciéndola significativamente, el lodo activado se separa del agua tratada y otra parte se recicla de vuelta al proceso, manteniendo una alta concentración de microorganismos para la degradación continua de la materia orgánica, y el resto se elimina como exceso de lodo.

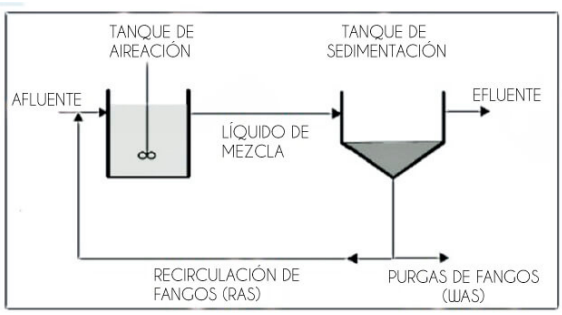

Figura 9: Proceso de fangos activados [24].

4.3. Plataforma de simulación BSM1

La Plataforma de simulación Benchmark Simulation Model No. 1 (BSM1) es una herramienta utilizada para modelar y evaluar procesos de tratamiento de aguas residuales, en particular el proceso de fangos activados. Incluye la representación de cinco tanques interconectados: el tanque de pretratamiento, el reactor biológico, el clarificador secundario, el tanque de reciclado de lodos y el tanque de desecho de lodos.

En el tanque de pretratamiento, las aguas residuales se preparan para el proceso de tratamiento. Esto suele implicar la eliminación de sólidos grandes y otros materiales que podrían interferir con el tratamiento posterior. Después del pretratamiento, el agua residual fluye hacia el reactor biológico. Aquí se mezcla con el fango activado, que contiene una alta concentración de microorganismos. Este tanque se airea para fomentar la actividad microbiana, que descompone la materia orgánica en el agua residual.

Luego, el agua y los lodos activados se trasladan al clarificador secundario, donde el lodo se separa del agua tratada por sedimentación. Una parte del lodo separado se recicla de vuelta al reactor biológico a través del tanque de reciclado de lodos, para mantener una alta concentración de microorganismos y asegurar la continuidad del proceso. El lodo restante se dirige al tanque de desecho de lodos para su eliminación final, como se puede observer en la figura 10. El BSM1 permite modelar y simular todos estos pasos de manera efectiva, lo que facilita la optimización del tratamiento de aguas residuales.

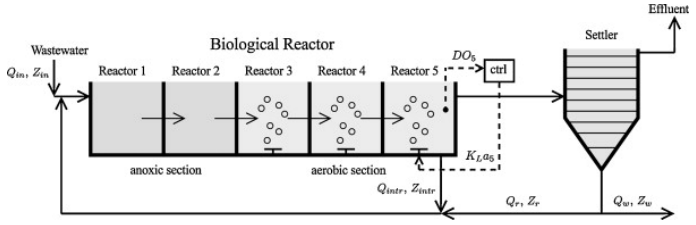

Figura 10: Visión general de la planta BSM1 [2].

El modelo BSM1 no tiene en consideración a la dinámica hidráulica de la plan-

ta ya que los reactores se encuentran conectados en forma de cascada lo que se representa en la ecuación 5:

$$Q_1 = Q_{in} + Q_{intr} + Q_r \tag{5}$$

Donde $Q_{intr} = 55,338[m^3d^{-1}]$ es la tasa de reciclado interno de nitratos del quinto reactor al primer reactor, $Q_r = 18,466[m^3d^{-1}]$ es la tasa de reciclado de lodos del flujo inferior del decantador secundario a la parte delantera de la planta, y Q_{in} es el caudal afluente con un valor medio igual a $18,466[m^3d^{-1}]$ [2]. Por último el decantador secundario se lo considera una unidad que no tiene reacción biológica con el volumen de $6000[m^3]$, por lo que para este modelo se tiene que subdividir para 10 capas y tomar en cuenta las transferencias verticales entre capas, considerando que la sexta capa es la de alimentación.

4.4. Modelo de planta

Para la descripción de los fenómenos biológicos se ha seleccionado el Modelo ASM1, el cual tiene lugar en los reactores biológicos que se encontraron en la figura 8, Se identifica que el modelo mecánico dentro del cual se identifican los componentes que se consideran de relevancia y a su vez los procesos biológicos más importantes dentro del modelo. El ASM1 cuenta con alrededor de 13 variables en las que se definirán a continuación en la tabla 4.4.

Definición	Notación	Fuente
Materia orgánica inerte soluble	S_I	[2]
Sustrato fácilmente biodegradable	S_S	[2]
Materia orgánica inerte particulada	X_I	[2]
Sustrato lentamente biodegradable	X_S	[2]
Biomasa heterótrofa activa	$X_{B,H}$	[2]
Biomasa autótrofa activa	$X_{B,A}$	[2]
Productos particulados derivados de la descomposición de la biomasa	X_P	[2]
Oxígeno	DO	[2]
Nitrógeno nitrato y nitrito	S_{NO}	[2]
Nitrógeno $NH_4^+ + NH_3$	S_{NH}	[2]
Nitrógeno orgánico biodegradable soluble	S_{ND}	[2]
Nitrógeno orgánico biodegradable particulado	X_{ND}	[2]
Alcalinidad	S_{ALK}	[2]

Por cada variable se considera que se encuentra asociada con una tasa de conversión lo que resulta la combinación de 8 procesos básicos que dependen de algunas variables como en estado no lineal, es así que la tasa de conversión de oxígeno viene dada por la ecuación a continuación:

$$r(DO) = -\frac{1 - Y_H}{Y_H}\rho_1 - \frac{4{,}57 - Y_A}{Y_A}\rho_3 \tag{6}$$

donde ρ_1 es el aumento aeróbico de heterótrofos modelado como:

$$\rho_1 = \mu_H \left(\frac{S_S}{K_S + S_S} \right) \left(\frac{DO}{K_{0,H} + DO} \right) X_{B,H} \tag{7}$$

y ρ_3 es el aumento aeróbico de autótrofos modelado como:

$$\rho_3 = \mu_A \left(\frac{S_{NH}}{K_{NH} + S_{NH}} \right) \left(\frac{S_O}{K_{0,A} + DO} \right) X_{B,A} \tag{8}$$

donde Y_H y Y_A son los parámetros estequiométricos, y μ_H, K_S, $K_{O,H}$, μ_A, K_{NH} y $K_{O,A}$ son los parámetros cinéticos. S_S es la materia orgánica inerte soluble, $X_{B,H}$ es la biomasa heterótrofa activa, S_{NH} es el nitrógeno $NH_4^+ + NH_3$ y $X_{B,A}$ es la biomasa autótrofa activa [2].

4.5. Perturbaciones en el influente

Las perturbaciones que se utilizan para determinar las estrategias del control, tienen un papel muy importante dentro de la evaluación ya que los lodos activados cuentan con una naturaleza polifacética, es por ello que la estrategia a elegir debe tener el control de accionamiento con una perturbación y por otro lado no de la misma manera con la otra. En ese caso para poder realizar una evaluación correcta y de forma imparcial, se necesitan varias perturbaciones que se van a someter a diversas estrategias de control. Para el modelo de BSM1, se definen 3 perturbaciones del influente que son representativas dentro del medio meteorológico, los cuales son: tiempo seco, lluvioso y por último el tiempo tormentoso. Dentro de cada escenario se manejan 28 días de recolección de datos de influente en intervalos de muestras de 15 min, en este caso se evalúan 14 días de datos dinámicos.

5. Descripción del Caso de Estudio

Recurrir a una fuente de información y analizarla es un parte primordial de un proyecto, ya que con esto se puede tomar decisiones importantes sobre un proceso en específico. En esta sección del proyecto se presenta una síntesis general de los escenarios de control sobre los cuales se simuló la planta y las variables asociadas. Bajo estas condiciones fue posible obtener un volumen importante de datos, que posteriormente fueron almacenados en un dataset y finalmente analizados mediante distintos métodos estadísticos, entre los cuales figuran: la t-Studen, coeficiente de correlación de Pearson y coeficiente de determinación R^2; todos con el fin de determinar, entre otros, si las variables del dataset presentan algún tipo de relación entre ellas.

5.1. Obtención de datos de la plataforma de simulación BSM1

Con el objetivo de realizar una comparación del funcionamiento de la planta de tratamiento de aguas residuales (Waster Water Treament Plan – WWTP) se propusieron dos escenarios de control: el primero, en donde la planta funciona a lazo abierto (es decir, no existe control sobre la misma) y el segundo, en donde se usa un modelo de control por defecto del BSM1 basado en PID's. Ambos escenarios fueron simulados a través del modelo estándar Benchmark N°1-BSM1 el cual basa su funcionamiento en el modelo de lodos activados No.1 (ASM1) desarrollado por la Asociación Internacional para la Investigación y el Control de la Contaminación del Agua [28].

La planta es un sistema multivariable (ver figura 14) con gran interacción, las variables consideradas en el entrenamiento de las redes se muestran en la tabla 1.

Tabla 1: Variables tomadas de la simulación de la planta

Descripción	Variable	Fuente
S_{NH}	Concentración de nitrógeno en forma de $NH_4^+ + NH_3$ $[gN/m^3]$	[29]
S_{NO}	Concentración de nitrógeno en forma de nitratos y nitritos $[gN/m^3]$	[29]
S_{ND}	Concentración de nitrógeno orgánico soluble biodegradable $[gN/m^3]$	[29]
Q	Tasa de Flujo $[m^3/d]$	[29]
$S_{NH} \cdot Q$	Concentración de nitrógeno en forma de $NH_4^+ + NH_3$ por Tasa de Flujo $[gN/d]$	[29]
$S_{NO} \cdot Q$	Concentración de nitrógeno en forma de nitratos y nitritos por Tasa de Flujo $[gN/d]$	[29]
$S_{ND} \cdot Q$	Concentración de nitrógeno orgánico soluble biodegradable por Tasa de Flujo $[gN/d]$	[29]

Para cada escenario de control se tomatón en cuenta tres tipos de influente que representan tres condiciones meteorológicas: clima seco o DRY, clima lluvioso o RAIN (combinación de clima seco y un periodo largo de lluvia) y clima tormentoso o STORM (combinación de clima seco y tormentas). Cada escenario se simuló por 14 días con un tiempo de muestreo de 15 minutos.

Puesto que la planta de tratamiento cuenta con varios componentes, con el objetivo de realizar un análisis más detallado de la misma, para cada escenario de control y a para cada condición meteorológica se midieron las variables de la Tabla 1 en tres puntos de referencia, tal como se detalla a continuación:

- **Punto 1:** Influente (ingreso del agua contaminada a la planta).
- **Punto 2:** Ingreso del tanque 1 (reactor biológico anóxico), en donde se lleva a cabo el proceso de desnitrificación.

- **Punto 3:** Salida del tanque 5 (reactor biológico aerobio), en donde se lleva a cabo el proceso de nitrificación.

Finalmente, con las consideraciones mencionadas, de cada simulación se extrajeron los valores de las variables de la Tabla 1 para formar una base de datos, y posteriormente, realizar un análisis estadístico.

5.2. Métodos estadísticos propuestos

De acuerdo a la base de datos obtenida, se llevaron a cabo tres tipos de análisis estadísticos: coeficiente de correlación de Pearson, coeficiente de determinación R^2 y t-Student.

El objetivo de estos análisis es determinar si los componentes químicos residuales en el líquido efluente tienen algún tipo de relación con los químicos residuales medidos en cada uno de los tres puntos definidos para la planta. Los resultados obtenidos se muestran y explican a continuación.

5.3. Análisis de los resultados usando Coeficiente de correlación de Pearson

La correlación de Pearson r_{xy} es una herramienta estadística que generalmente se usa como prueba de hipótesis y sirve para determinar si existe o no relación lineal entre dos variables que se están comparando. El coeficiente de correlación de Pearson puede tomar valores entre -1 y $1(-1 \leq r_{xy} \geq -1)$, en donde, si el valor de r_{xy} se acerca a 1 entonces las variables tendrán una relación directa (es decir si aumenta la una, la otra también lo hace); si por el contrario, el valor de r_{xy} es cercano a -1 entonces la relación es inversa. En caso de que el valor de r_{xy} se acerque a cero, entonces las variables no estarán relacionadas linealmente [4].

Para nuestro caso de estudió, se determinó el coeficiente de correlación de Pearson de cada una de las variables de la Tabla 1 respecto la variable S_{NH} del efluente para cada condición meteorológica (Dry, Rain y Storm) y se ordenaron tal como se detalla en la Tabla 2.

Tabla 2: Detalle de cálculo para coeficiente de correlación de Pearson

DRY	S_{NH} * $S_{NH}E$	S_{NO} * $S_{NH}E$	S_{ND} * $S_{NH}E$	Q * $S_{NH}E$	$S_{NH} \cdot Q$ * $S_{NH}E$	$S_{NO} \cdot Q$ * $S_{NH}E$	$S_{ND} \cdot Q$ * $S_{NH}E$
RAIN	S_{NH} * $S_{NH}E$	S_{NO} * $S_{NH}E$	S_{ND} * $S_{NH}E$	Q * $S_{NH}E$	$S_{NH} \cdot Q$ * $S_{NH}E$	$S_{NO} \cdot Q$ * $S_{NH}E$	$S_{ND} \cdot Q$ * $S_{NH}E$
STORM	S_{NH} * $S_{NH}E$	S_{NO} * $S_{NH}E$	S_{ND} * $S_{NH}E$	Q * $S_{NH}E$	$S_{NH} \cdot Q$ * $S_{NH}E$	$S_{NO} \cdot Q$ * $S_{NH}E$	$S_{ND} \cdot Q$ * $S_{NH}E$
	S_{NH}	S_{NO}	S_{ND}	Q	$S_{NH} \cdot Q$	$S_{NO} \cdot Q$	$S_{ND} \cdot Q$

A continuación de analizan los resultados determinados:

5.3.1. Correlación de Pearson – Punto 1(Influente)

Al observar la Fig. 11, se puede inferir que en general, para ambos escenarios de control, el coeficiente de correlación de Pearson r_{xy} presenta valores muy cercanos a cero, lo que indica que las variables relacionadas presentan una correlación lineal muy débil o nula. Es decir, que los componentes químicos medidos en el punto influente de la planta no se relacionan con el S_{NH} del efluente. El elemento que parece destacar es la concentración de nitrógeno soluble (S_{ND}) del punto influente, ya que al observar la Fig. 11 en la estación Dry de lazo cerrado, el SND alcanza un valor máximo de 0.35 indicando una relación moderada con el S_{NH} del efluente.

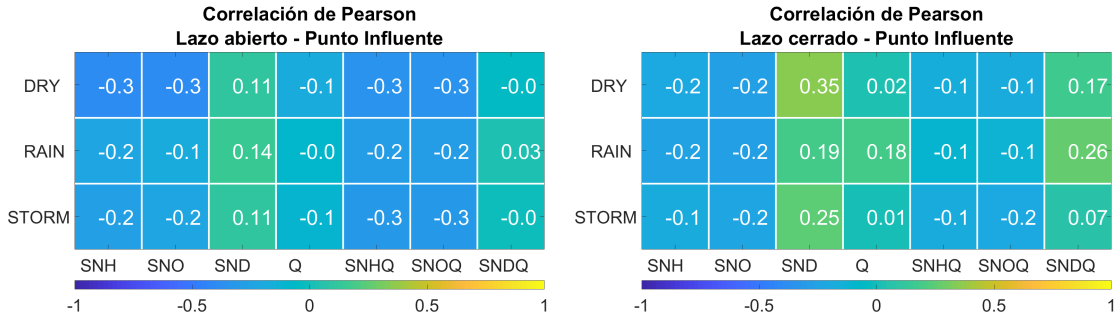

Figura 11: Coeficiente de correlación de Pearson – Punto 1(Influente).

5.3.2. Correlación de Pearson – Punto 2 (Tanque 1)

Para el Punto 2 (Tanque 1), la Fig. 12 Coeficiente de correlación de Pearson – Punto 2 (Tanque 1 Fig. 12 muestra que para ambos escenarios de control (lazo abierto y control MPC) el coeficiente de correlación de Pearson presenta un nivel de asociación de variables moderado. Un ejemplo es la cantidad de nitrógeno soluble biodegradable $S_{ND} \cdot Q$ del escenario de lazo cerrado (para la estación meteorológica Rain), el cual presenta una relación lineal directa (de 0.57) con el SND del efluente.

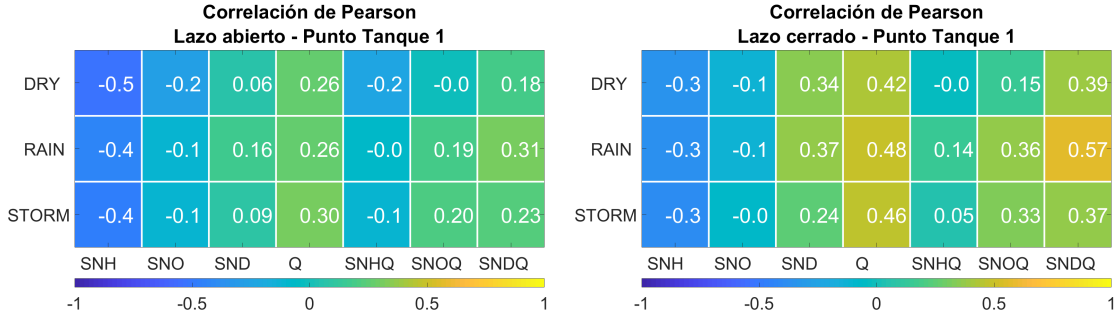

Figura 12: Coeficiente de correlación de Pearson – Punto 2 (Tanque 1).

5.3.3. Correlación de Pearson – Punto 3 (Tanque 5)

En el Punto 3 (Tanque 5) de la planta se observa que, en ambos escenarios, todas las variables tienen una relación directa la concentración de nitrógeno del efluente (Ver Fig. 13). En este caso el nivel de correlación lineal es directo y muy fuerte (cercano a 1). Tal es el caso para la concentración de nitrógeno orgánico biodegradable (S_{ND}) para la estación Rain en el escenario de lazo cerrado (Fig. 13 - derecha) que presenta un valor de 0.78. Estos resultados son lógicos puesto que el punto 3 (tanque 5) es el punto que está más cerca del efluente de la planta, por lo que se podría intuir que la cantidad de nitrógeno orgánico SND medido en este punto tendrá un valor similar al del punto emergente de la planta.

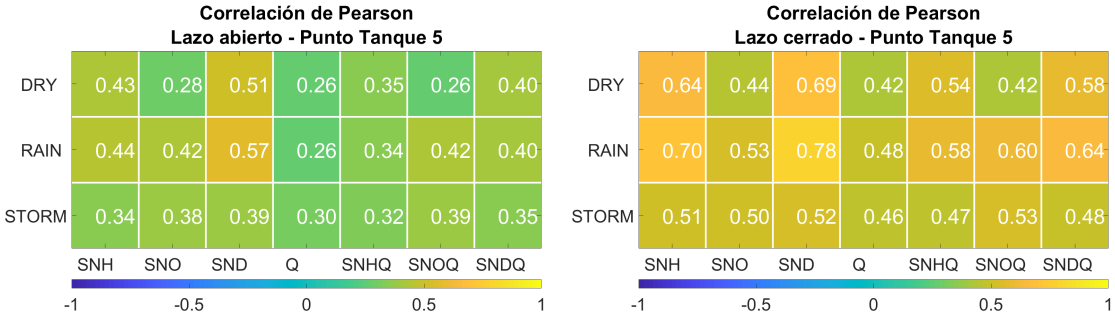

Figura 13: Coeficiente de correlación de Pearson – Punto 3 (Tanque 5).

Como comentario general, se puede mencionar que para los tres análisis estadísticos presentados hasta ahora (análisis para los puntos influentes, tanque 1 y tanque 5), la concentración de nitrógeno orgánico soluble biodegradable (S_{ND}) es la variable que tiene la relación más fuerte con el S_{NH} del efluente. Esto puede explicarse al observar la Fig. 14; en esta figura se presentan los procesos biológicos que caracterizan al modelo ASM1 y las variables asociadas cada uno. Como se observa, las variables S_{ND} y S_{NH} tienen una estrecha relación a través del proceso biológico de amonificación, en el cual, básicamente, el nitrógeno orgánico soluble se transforma en amonio $NH4^+$ [34].

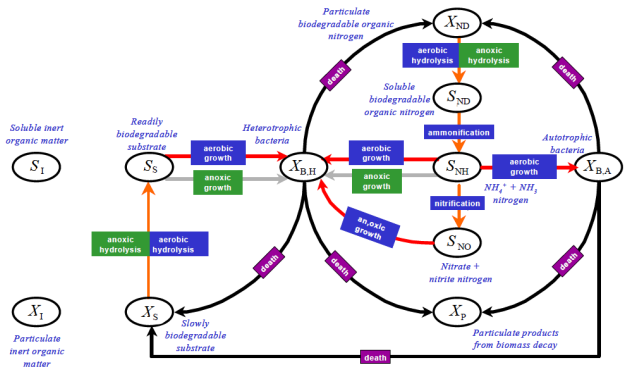

Figura 14: Procesos biológicos reflejados en al ASM1 [1].

5.4. Análisis de los resultados usando Coeficiente de determinación R^2

A diferencia del coeficiente de correlación de Pearson que mide la relación de linealidad de dos variables, el coeficiente de determinación R^2 tiene un objetivo diferente, y es medir lo bien que un modelo se ajusta a los datos reales observados. El coeficiente de determinación R^2 es una medida acotada, siendo sus límites $0 < R^2 < 1$. Un R^2 de uno (1) significa un ajuste perfecto, y un valor cero (0) indica la no representatividad del modelo, lo que significa que el modelo usado no explica nada de la variación de los datos reales observados [25].

Para esta parte de nuestro caso de estudió se consideró el coeficiente de determinación R^2 de cada una de las variables de la Tabla 1 respecto la concentración de nitrógeno en forma de $NH_4^+ + NH_3$ del efluente para cada condición climática (Dry, Rain y Storm) y se ordenaron tal como se detalla en la Tabla 3.

Tabla 3: Detalle de cálculo para coeficiente de determinación R^2.

DRY	$R^2(S_{NH} \cdot S_{NH}E)$	$R^2(S_{NO} \cdot S_{NH}E)$	$R^2(S_{ND} \cdot S_{NH}E)$	$R^2(Q \cdot S_{NH}E)$
RAIN	$R^2(S_{NH} \cdot S_{NH}E)$	$R^2(S_{NO} \cdot S_{NH}E)$	$R^2(S_{ND} \cdot S_{NH}E)$	$R^2(Q \cdot S_{NH}E)$
STORM	$R^2(S_{NH} \cdot S_{NH}E)$	$R^2(S_{NO} \cdot S_{NH}E)$	$R^2(S_{ND} \cdot S_{NH}E)$	$R^2(Q \cdot S_{NH}E)$
	S_{NH}	S_{NO}	S_{ND}	Q

Los resultados y análisis de este cálculo estadístico se presentan a continuación:

5.4.1. Coeficiente de determinación R^2 – Punto 1(Influente)

Como se puede observar en la Fig. 15, para el punto 1 (en ambos escenarios de control), ninguna de las variables (S_{NH}, S_{NO}, S_{ND} y Q) se ajusta a la tendencia dada por la S_{NH} del efluente. Los valores determinados para el coeficiente de determinación R^2 es casi nulo en todos los casos.

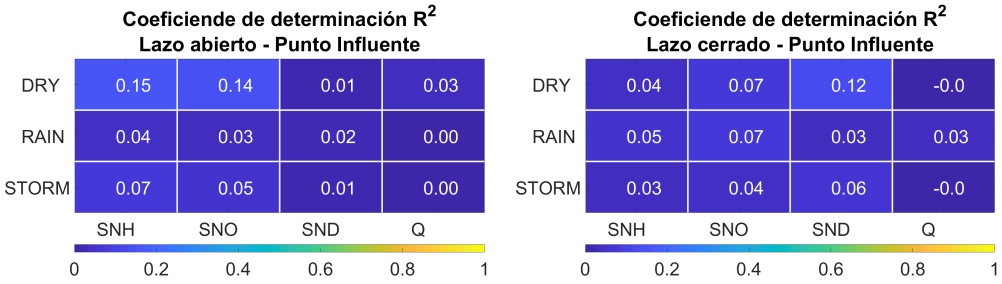

Figura 15: Coeficiente de determinación R^2 – Punto 1(Influente).

5.4.2. Coeficiente de determinación R^2 – Punto 2 (Tanque 1)

En el caso del punto 2 (tanque 1), se aprecia que en la Fig. 16, al igual que en el punto 1, el coeficiente de determinación presenta valores cercanos a cero para ambos escenarios de control (lazo abierto y lazo cerrado). El máximo valor encontrado es el 0.28 para la concentración de nitrógeno (S_{NH}) respecto al S_{NH} del efluente. Aún con este hecho, se puede afirmar que, para el punto 2, las variables S_{NH}, S_{NO}, S_{ND} y Q no se ajustan al modelo dado por el S_{NH} del efluente.

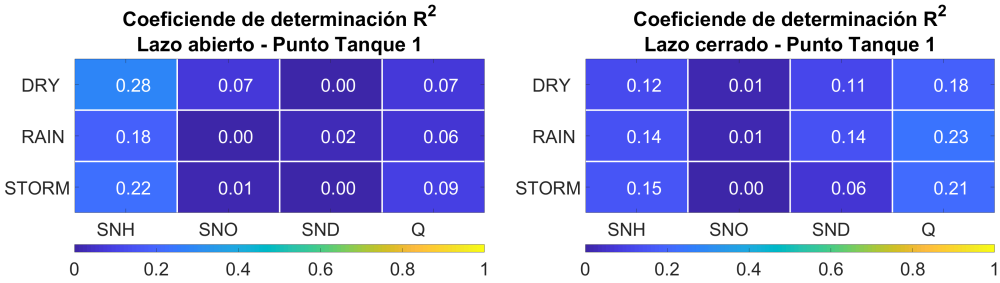

Figura 16: Coeficiente de determinación R^2 – Punto 2 (Tanque 1).

5.4.3. Coeficiente de determinación R^2 – Punto 3 (Tanque 5)

Tal como se observa en la Fig. 17, el coeficiente de determinación para el escenario de lazo abierto demuestra que las variables S_{NH}, S_{NO}, S_{ND} y Q no se ajustan al comportamiento de la concentración de nitrógeno S_{NH} del efluente. Para el escenario de lazo cerrado (Fig. 17-derecha) la única variable que presenta un nivel de ajuste medio es la concentración de nitrógeno orgánico biodegradable (S_{ND}) con un valor 0.61 (61 %) para la estación Rain.

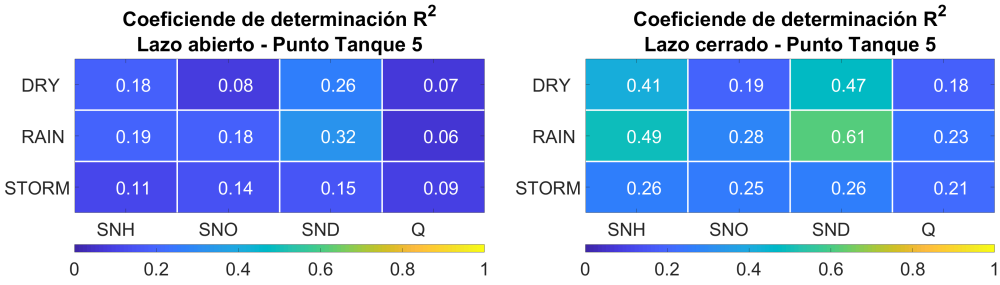

Figura 17: Coeficiente de determinación R^2 – Punto 3 (Tanque 5).

5.5. Análisis de los resultados usando la t-Student

La contribución de la prueba t-Student, específicamente, es para comparar dos grupos de datos o muestras. Al ser una prueba de hipótesis, la hipótesis nula establece que no hay diferencia entre las medias de los grupos de datos, mientras que

la hipótesis alternativa establece que, de existir una diferencia, esta solo se debe al azar [27].

Para nuestro caso de estudió, se realizó la comparación de las variables S_{NH}, Q y S_{NO} con el S_{NH} del efluente para los tres puntos elegidos en la planta (en cada condición meteorológica y en escenario de lazo cerrado) usando la prueba t-Student. Estos resultados se muestran en la Tabla 4, Tabla 5 y Tabla 6.

Tabla 4: Resultados prueba t-Student en lazo cerrado para el punto 1 (Influente).

h del Punto 1 – Influente			
DRY	1	1	1
RAIN	1	1	1
STORM	1	1	1
	S_{NH}	Q	S_{NO}

Tabla 5: Resultados prueba t-Student en lazo cerrado para el punto 2 (tanque 1).

h del Punto 2 – Influente			
DRY	1	1	1
RAIN	1	1	1
STORM	1	1	1
	S_{NH}	Q	S_{NO}

Tabla 6: Resultados prueba t-Student en lazo cerrado para el punto 3 (Tanque 5).

h del Punto 3 – Influente			
DRY	1	1	1
RAIN	1	1	1
STORM	1	1	1
	S_{NH}	Q	S_{NO}

Los valores observados en las columnas de las tres tablas son el resultado de la prueba de hipótesis t-Student (representado con la variable h). Si el valor de h es 1 indica que la prueba t-Student rechaza la hipótesis nula a un nivel de significación del 5 %. Es decir, para la planta en estudió, en todos los casos, la concentración de nitrógeno en forma de nitratos y nitritos (S_{NO}), la concentración de nitrógeno en forma de NH4++NH3 (S_{NH}) y la concentración de nitrógeno orgánico soluble biodegradable (S_{ND}) en los puntos 1, 2 y 3, no es igual a la concentración de nitrógeno en forma de $NH_4^+ + NH_3$ (S_{NH}) medida en el efluente.

5.6. Limpieza de Datos

Del análisis realizado al dataset recolectado para este estudio, podemos observar que los valores representados en los diferentes campos muestran precisión y consistencia sin que exista algún tipo de anormalidad o datos no coherentes que afecten a

este estudio. Además, se observa que el formato y la estructura expuesta no muestran errores, y que los datos obtenidos para cada variable están dentro de rangos esperados de acuerdo con el dominio del problema.

Es importante destacar que la fuente de los datos es confiable y tuvo un correcto manejo en cada simulación, por lo que se sugiere que se incorpore un control de calidad para reducir la probabilidad de error. Con esto, se puede concluir que los datos obtenidos para este estudio cumplen con los requisitos y formatos requeridos para ser aplicados en el BSM1 de simulink, es decir, no es necesario realizar una limpieza de datos adicional, por tanto, se demuestra que el dataset a utilizar dentro de esta investigación es adecuado y no requiere de modificaciones significativas.

6. Diseño de la Red Neuronal

La ciencia de datos, el machine learning y las redes de aprendizaje profundo o Deep learning en términos técnicos ha demostrado ser de gran ayuda para la toma de decisiones basadas en los datos y no en la intuición en diferentes campos como la industria, la academia y la medicina. Los datos históricos de serie cronológica desempeñan un papel importante en muchas implementaciones, lo que muestra la capacidad que tienen las redes neuronales en tratar este tipo de información.

En el marco de esta investigación de máster, se tomo la decision de explorar el uso de dos tipos principales de redes neuronales: las redes perceptrón multicapa (MLP) y las redes de gran memoria a corto plazo (LSTM). Estas dos arquitecturas han demostrado ser muy eficaces en tareas de predicción y modelado, y cada una tiene sus propias ventajas y características únicas que las hacen adecuadas para distintas clases de problemas.

La red MLP, posee una alimentación hacia adelante, es conocidas por su capacidad para modelar funciones no lineales y su facilidad de uso asi como de ser fácilmente escalable a grandes conjuntos de datos y ser altamente personalizable en términos del número de capas ocultas y unidades en cada capa. Mientras que la LSTM es un tipo de red recurrente que ha sido diseñada para abordar el problema de la desaparición del gradiente en el entrenamiento de las RNNs. La ventaja que la caracteriza es la capacidad para aprender dependencias a corto plazo, lo que la hace muy adecuada para tareas que involucran datos secuenciales o temporales.

Se exploraran y compararan ambas arquitecturas de redes en el contexto del control de sistemas de tratamiento de aguas residuales, con el objetivo de desarrollar y optimizar controladores inteligentes capaces de predecir y regular eficazmente el comportamiento de estos sistemas. A continuación, se detalla el diseño y el procedimiento de entrenamiento de cada uno de estos modelos.

6.1. Multi Layer Perceptron (MLP)

6.1.1. Red para datos obtenidos del sistema de control en lazo abierto

6.1.1.1. Diseño y optimización de la red MLP

Inicia con una capa de entrada de secuencia que recibe los datos normalizados de entrada. Luego, tiene cuatro capas ocultas, cada una con 160 unidades ocultas. Después de cada capa oculta, se aplica una capa de activación ReLU para introducir no linealidades y una capa de dropout con una tasa de 0.2 para evitar el sobreajuste. La red concluye con una capa completamente conectada que se conecta a una capa de regresión, que se utiliza para predecir el valor de salida, como se puede observar en la figura 18.

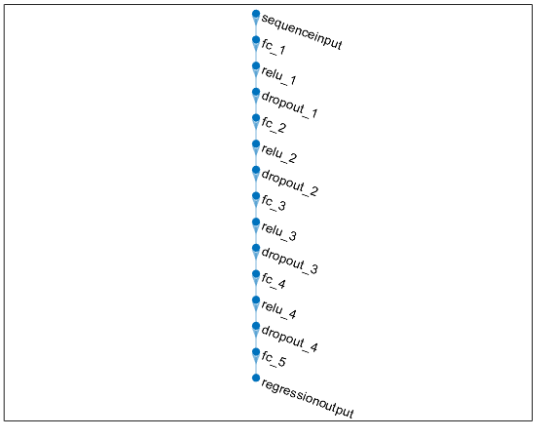

Figura 18: Estructura de la red neuronal MLP.

La optimización de esta red se lleva a cabo mediante el optimizador Adam. Este método es conocido por su eficacia en la resolución de problemas no convexos asociados con el entrenamiento de redes neuronales. La tasa de aprendizaje se ajusta de acuerdo con un enfoque "piecewise", donde se reduce en un factor de 0.5 cada 200 épocas.

6.1.1.2. Entrenamiento y configuración de la red MLP

El conjunto de datos utilizado para el entrenamiento de la red se divide en tres categorías basadas en las condiciones climáticas: seco, lluvia, y tormenta. Los datos se normalizan en el rango de 0 a 1 antes del entrenamiento para garantizar una convergencia más rápida y un mejor rendimiento de la red. Para cada categoría, los últimos 16 registros equivalentes a 4 horas, ya que cada registro representa 15 minutos, se omiten del conjunto de entrenamiento y se usan como objetivo para predecir el comportamiento futuro del sistema.

Los hiperparámetros son valores establecidos antes de iniciar el entrenamiento y no se actualizan durante el mismo. Estos incluyen aspectos como la tasa de aprendizaje, el número de capas ocultas en la red, el número de unidades en cada capa, entre otros. La tabla 7 presenta los hiperparámetros utilizados para configurar y entrenar la red neuronal MLP en MATLAB, estos valores han sido seleccionados y ajustados para proporcionar el mejor rendimiento posible con nuestros datos de entrada y salida.

Tabla 7: Hiperparámetros MLP para lazo abierto.

Hiperparámetro	**Valor utilizado**
Unidades de entrada	4
Unidades ocultas	160
Número de clases	1
Número de capas ocultas	3
Función de activación	ReLU
Dropout	0.2
Tamaño de lote (mini-batch)	256
Máximo número de épocas	1000
Frecuencia de validación	10
Aprendizaje programado (LearnRateSchedule)	piecewise
Factor de caída del aprendizaje (LearnRateDropFactor)	0.5
Periodo de caída del aprendizaje (LearnRateDropPeriod)	200

Además de la configuración de los hiperparámetros, otro aspecto importante es el seguimiento del progreso. La figura 19 proporciona información en tiempo real sobre la evolucionando del modelo a lo largo de las iteraciones de entrenamiento, incluyendo métricas como el error de entrenamiento y validación. Esto nos permite detectar problemas como el sobre ajuste y tomar decisiones sobre cuándo detener el entrenamiento.

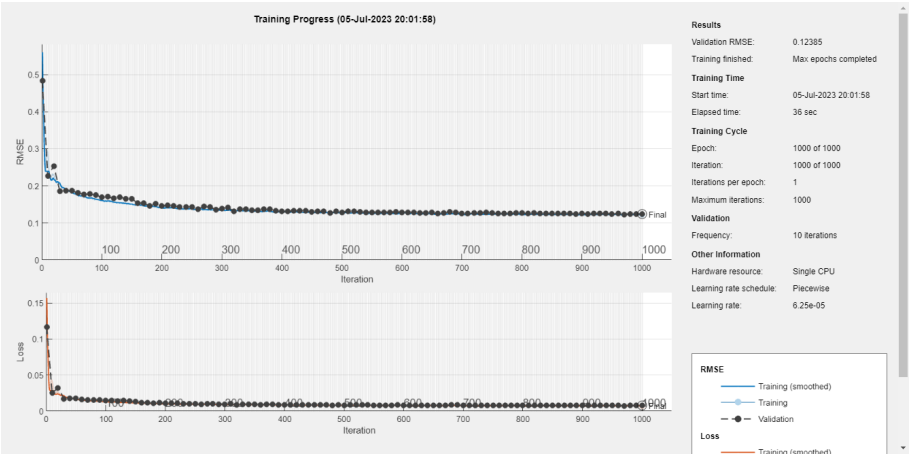

Figura 19: Progreso de entrenamiento MLP.

En la figura 19 se establece un hardware de un solo CPU para realizar el entrenamiento, lo que puede afectar la velocidad a la que se procesa cada época. No obstante,

un solo CPU puede ser suficiente para el entrenamiento de redes neuronales, especialmente si el conjunto de datos no es extremadamente grande. El aprendizaje se ajusta según la tasa de aprendizaje ”piecewise”, lo que significa que se reduce por un factor después de un número específico de épocas. En este caso, se estableció una tasa de aprendizaje inicial de 6.25e-0.5, ya que es fundamental determinar la velocidad a la que el modelo aprende. Una tasa de aprendizaje demasiado alta puede hacer que el modelo oscile alrededor de la solución óptima, mientras que una tasa de aprendizaje demasiado baja puede hacer que el aprendizaje sea muy lento o que el modelo se quede atascado en mínimos locales.

Se debe prestar atención en la pérdida de entrenamiento como a la pérdida de validación. Ambas deben disminuir a lo largo del tiempo y converger a valores similares. Si la función de perdida de validación comienza a aumentar, mientras que la pérdida de entrenamiento sigue disminuyendo, es posible que estemos ante un caso de sobre ajuste. En este escenario, el modelo está aprendiendo las características del conjunto de entrenamiento sin ningún problema, pero no es capaz de generalizar eficazmente a datos no vistos.

6.1.2. Red para datos obtenidos del sistema de control en lazo cerrado

6.1.2.1. Diseño y optimización de la red MLP

La arquitectura de la red es de naturaleza profunda, compuesta por una capa de entrada de secuencia y cuatro capas ocultas. Cada una de las etapas ocultas está constituida por unidades completamente conectadas, seguidas de una capa de activación ReLU para incorporar no linealidades en el modelo y así permitir que la red aprenda y modele relaciones más esenciales entre los datos. Para prevenir el sobre ajuste, se incluye una capa de dropout después de cada capa ReLU. Este diseño bien estructurado, se puede observar en la figura 20, el cual busca alcanzar una capacidad de modelado, manteniendo siempre un control sobre el posible sobre ajuste y garantizando una buena capacidad de generalización.

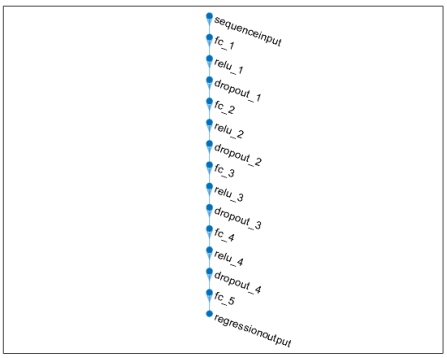

Figura 20: Estructura de la red neuronal MLP.

Para la optimización se emplea el algoritmo de Adam, una modalidad de descenso de gradiente estocástico reconocida por su eficacia en el entrenamiento de redes profundas. Se utilizan estrategias avanzadas de programación de la tasa de aprendizaje, incluyendo una reducción "piecewise" que ajusta la tasa de aprendizaje en ciertos intervalos. Este enfoque proporciona un equilibrio entre convergencia inicial y la optimización de largo plazo, garantizando que el modelo resultante sea robusto y capaz de adaptarse a diversas situaciones de predicción, destacando su aplicabilidad y eficacia en el ámbito de estudio.

6.1.2.2. Entrenamiento y configuración de la red MLP

Los hiperparámetros seleccionados para el entrenamiento de la red pueden observarse en la tabla 8, proporciona una visión clara de la configuración utilizada en el entrenamiento, desde la cantidad de características de entrada hasta los parámetros específicos del entrenamiento como número de épocas, tamaño del lote y frecuencia de validación. La configuración detallada de los hiperparámetros proporciona una base para realizar los experimentos o llevar acabo ajustes futuros con el fin de mejorar la precisión del modelo.

Tabla 8: Hiperparámetros MLP para lazo cerrado.

Hiperparámetro	**Valor**
Número de características	Basado en los datos de entrada
Unidades ocultas	165
Número de clases	1
Número de capas ocultas	4
Dropout	0.2
Tamaño de lote	256
Máximo número de épocas	1000
Frecuencia de validación	10
Aprendizaje programado	piecewise
Factor de caída del aprendizaje	0.5
Periodo de caída del aprendizaje	200

Para el entrenamiento de la red, se utilizó una función de activación ReLU y un dropout de 0.2. Las unidades ocultas y las capas ocultas se fijaron en 165, y 4, respectivamente. Se entrenó con un tamaño de lote de 256, un máximo de 1000 épocas y una frecuencia de validación de cada 10 épocas. El aprendizaje se programó de manera "piecewise", con un factor de caída de 0.5 y un periodo de caída de 200. En la figura que ilustra el progreso de entrenamiento de nuestro modelo. La figura 21 permite apreciar el comportamiento de la red neuronal durante las iteraciones de entrenamiento, mostrando cómo los valores de pérdida de entrenamiento y validación evolucionan con el paso de las épocas. Esta visualización es fundamental para entender cómo nuestro modelo está aprendiendo y permite identificar potenciales problemas como el sobre ajuste.

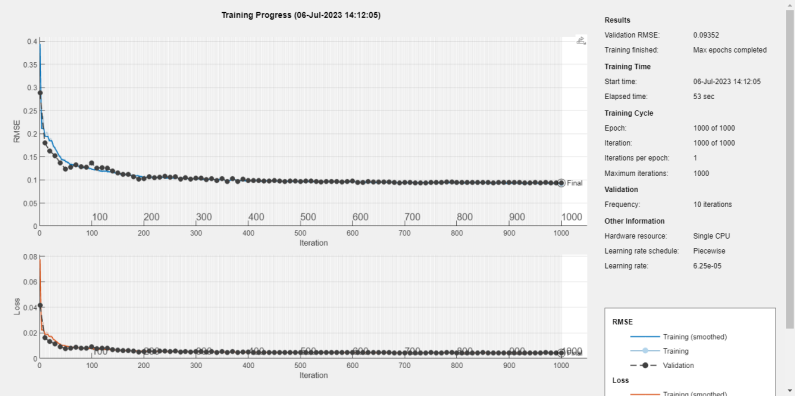

Figura 21: Progreso de entrenamiento MLP.

La figura 21 muestra una disminución constante en los valores de pérdida a medida que avanzan las iteraciones. Inicialmente, el modelo aprende a un ritmo bastante rápido, reflejado en la pronunciada caída de la pérdida durante las primeras 100 iteraciones, luego comienza a estabilizarse, mostrando un decrecimiento gradual en la pérdida. La tasa de aprendizaje utilizada es de 6.25e-05, es apropiada, ya que permite al modelo aprender de manera efectiva sin causar oscilaciones bruscas en la pérdida. Se observa que el modelo se valida cada 10 iteraciones, permitiendo tener un seguimiento frecuente del desempeño en el conjunto de validación. La pérdida de validación sigue una tendencia similar a la pérdida de entrenamiento, lo que nos indica que el modelo no está sobre ajustando los datos de entrenamiento, es decir, está aprendiendo de manera efectiva, con buen equilibrio entre velocidad de aprendizaje y prevención del sobre ajuste.

6.2. Long-Short Term Memory (LSTM)

6.2.1. Red para datos obtenidos del sistema de control en lazo abierto

6.2.1.1. Diseño y optimización de la red LSTM

El diseño empezó con una capa de entrada de secuencia que recibe los datos normalizados de entrada, seguido de dos capas ocultas, cada una compuesta por 50 unidades LSTM, luego de cada una de ellas, se aplica una capa de dropout con una tasa de 0.1 para mitigar el sobre ajuste, permitiendo que ciertas neuronas se acoplen durante el proceso de entrenamiento y así evitar que la red dependa de una pequeña cantidad de neuronas, se introduce una capa completamente conectada a una función de activación ReLU, con la finalidad de introducir la no linealidad en la red, permitiendo que aprenda y modele relaciones más complejas entre las entradas y las salidas.

El diseño de la red se muestra en la figura 22, donde se puede apreciar la secuencia de capas que constituyen la arquitectura de la red.

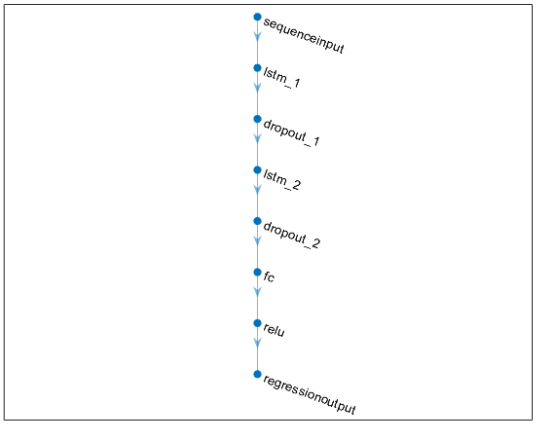

Figura 22: Estructura de la red neuronal LSTM.

Para optimizar el rendimiento, se aplicaron técnicas de ajuste de hiperparámetros. Se empleó el algoritmo Adam para el entrenamiento, configurando un número máximo de 400 épocas y un tamaño de minibatch de 64. Se usó un enfoque de validación durante el proceso de entrenamiento para evaluar el desempeño de la red, registrándose cada cinco épocas. Se optó por reorganizar los datos en cada época para evitar cualquier sesgo debido al orden de los datos. Con esto se buscó lograr un modelo que ofrezca precisión en las predicciones y resistencia al sobre ajuste.

6.2.1.2. Entrenamiento y configuración de la red LSTM

Los hiperparámetros utilizados para configurar y entrenar la red neuronal LSTM en MATLAB, han sido seleccionados y ajustados para proporcionar el rendimiento óptimo, este tipo de red neuronal se destaca por su capacidad para aprender dependencias a largo plazo, lo que la hace especialmente útil para tareas de secuencia a secuencia, como es el caso de nuestro estudio. La eficiencia y precisión de la LSTM reside en su arquitectura robusta y en el ajuste de los hiperparámetros que se presentan en la tabla 9.

Tabla 9: Hiperparámetros LSTM para lazo abierto.

Hiperparámetro	**Valor**
Épocas Máximas	400
Tamaño de Mini Lote	64
Frecuencia de Validación	5
Número de Capas Ocultas	2
Número de Unidades Ocultas	50
Tasa de Abandono (Dropout)	0.1
Tasa de Aprendizaje	Adam
Tipo de Capa de Entrada	Capa de Entrada de Secuencia
Tipo de Capa de Salida	Capa de Regresión
Función de Activación	ReLU
Método de Normalización	Escalado Min-Max

La red neuronal LSTM, construida y entrenada, presenta algunas diferencias clave en sus hiperparámetros con respecto a una red MLP. La cantidad de épocas y el tamaño del mini lote son dos de estos parámetros que se han ajustado de manera específica para la red. Se limitó las épocas máximas a 400, en comparación con las 1000 épocas utilizadas en la MLP. Esto se debe a que las redes LSTM son computacionalmente más costosas y se necesita encontrar un equilibrio entre el rendimiento y el tiempo de entrenamiento. Además, la red utiliza un tamaño de mini lote de 64, en lugar del tamaño de mini lote de 256 utilizados en la MLP, un tamaño más pequeño puede ayudar a aprender mejor las dependencias a largo plazo en los datos.

En cuanto al número de capas ocultas y las unidades ocultas, la red LSTM utiliza 2 capas ocultas con 50 unidades ocultas, mientras que la MLP utiliza 3 capas con 160 unidades. Aunque a primera vista la red LSTM parezca más simple, no es así, ya que cada unidad es un sistema complejo capaz de retener y olvidar información, por lo que su capacidad de modelado no se reduce. Es importante mencionar que el tiempo de entrenamiento de una red LSTM es mayor que el de una red MLP debido a su mayor complejidad. Sin embargo, la capacidad de la LSTM para modelar dependencias a largo plazo en los datos puede resultar en un mejor rendimiento en tareas de secuencia a secuencia.

La figura 23 muestra el progreso del entrenamiento de la red LSTM, el gráfico, muestra el comportamiento del error en los conjuntos de entrenamiento y validación a medida que avanza el número de épocas. La disminución del error en ambos conjuntos nos permite confirmar la efectividad del aprendizaje de la red y su capacidad para generalizar a datos no vistos.

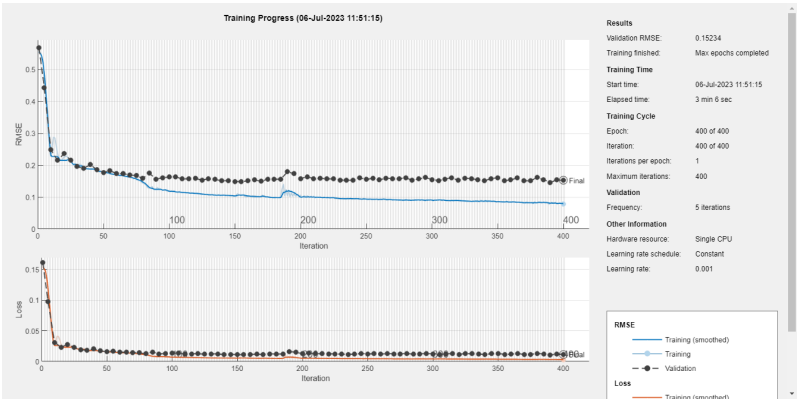

Figura 23: Progreso de entrenamiento LSTM.

En la figura 23 se observa una representación visual de cómo los valores de la raíz del error cuadrático medio y la función de pérdida evolucionan durante las iteraciones de entrenamiento, como era de esperarlo, existe una disminución constante en los errores tanto de entrenamiento como de validación, lo cual es un indicador de que nuestra red está aprendiendo y ajustándose adecuadamente con cada iteración del entrenamiento.

6.2.2. Red para datos obtenidos del sistema de control en lazo cerrado

6.2.2.1. Diseño y optimización de la red LSTM

En el diseño y optimización de la red, su configuración y arquitectura es fundamental, que a menudo implica hallar un equilibrio entre la complejidad y la capacidad de generalización. En esta investigacion, la red se compone de un nivel de entrada de secuencia, múltiples estratos LSTM ocultos y un nivel de regresión al final. A cada estrato LSTM le sigue un módulo de abandono con una tasa del 10 % para la regularización. Se emplea la función de activación ReLU en el nivel totalmente conectado para gestionar las no linealidades del modelo, permitiendo que la red capture y modele las relaciones más complejas entre los datos como se observa en la figura 24.

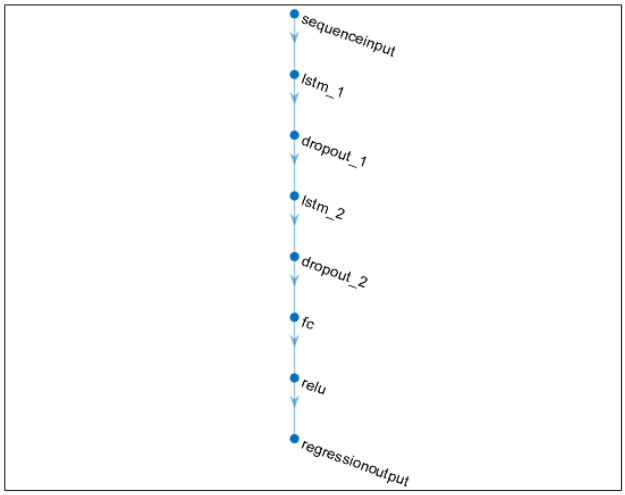

Figura 24: Estructura de la red neuronal LSTM.

En la optimización de la red, el número de unidades ocultas, el número de capas ocultas y la tasa de abandono son parámetros críticos. Con una adecuada selección y ajuste de estos parámetros. A través de la validación cruzada, la red puede ser entrenada para minimizar el error en los datos de entrenamiento mientras se mantiene una alta capacidad de generalización en los datos de validación y prueba. La eficacia del diseño y optimización de la red puede evaluarse mediante métricas de rendimiento como el error cuadrático medio, que proporciona una medida cuantitativa de la precisión del modelo.

6.2.2.2. Entrenamiento y configuración de la red LSTM

Los hiperparámetros configurados y entrenados para la red LSTM en MATLAB han sido elegidos y ajustados para obtener la mejor eficacia posible. Este tipo de

red es reconocido por su habilidad para aprender dependencias a largo plazo, una característica particularmente valiosa para tareas de secuencia a secuencia como la que se aborda en nuestro estudio. El desempeño preciso y eficaz de la red se debe a su sólida arquitectura y calibración de los hiperparámetros que se observan en la tabla 10.

Tabla 10: Hiperparámetros LSTM para lazo cerrado.

Hiperparámetro	**Valor**
Función de activación	ReLU
Dropout	0.1
Número de características	4 (dependiendo de los datos)
Número de unidades ocultas	50
Número de clases	1
Número de capas ocultas	2
Máximas épocas	400
Tamaño de lote miniatura	64
Frecuencia de validación	10
Agitación	en cada época
Programa de aprendizaje	”piecewise”
Factor de caída de aprendizaje	0.5
Periodo de caída de aprendizaje	200

En la figura 25, se muestra el progreso de entrenamiento para nuestra red, observando el rendimiento en los datos de entrenamiento como en los de validación, demostrando que la pérdida informacion del modelo afecta las predicciones y los valores reales, el eje vertical se representa la pérdida del modelo y en el horizontal, el número de épocas. Como se puede observar, ambas pérdidas disminuyen inicialmente, indicando que el modelo está aprendiendo a partir de los datos de entrenamiento. Sin embargo, a partir de la época 60, vemos una tendencia hacia la desestabilización del modelo. La pérdida en el conjunto de validación comienza a aumentar, lo cual es una señal de que el modelo puede estar empezando a sobreajustarse a los datos de entrenamiento generando una perdiendo en su capacidad para generalizar a nuevos datos.

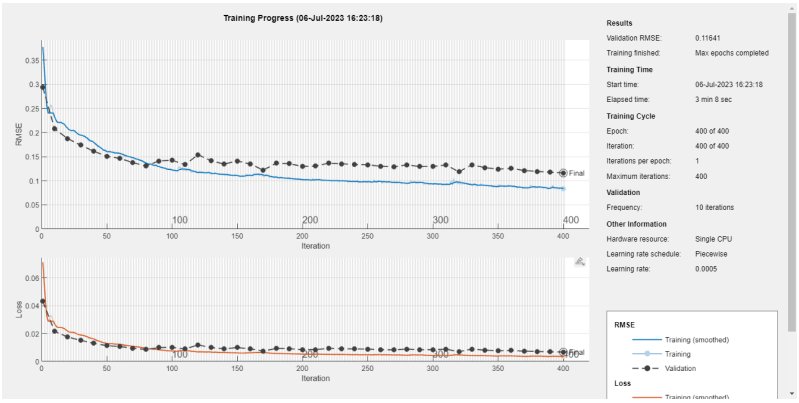

Figura 25: Progreso de entrenamiento LSTM.

7. Resultados y Discusión

Esta sección presenta las características comunes del entrenamiento en lazo abierto y cerrado de las redes neuronales MLP y LSTM y los resultados del entrenamiento y la validación en ciertos escenarios.

7.1. Comparación de desempeño en lazo abierto

Tabla 11: Hiperparámetros de las redes MLP y LSTM en modo lazo abierto

Hiperparámetro	**MLP**	**LSTM**
Número de características	4	4
Número de unidades ocultas	160	50
Número de clases	1	1
Número de capas ocultas	4	2
Tasa de abandono	0.2	0.1
Número máximo de épocas	1000	400
Tamaño del lote	256	64
Frecuencia de validación	10	5
Factor de caída de la tasa de aprendizaje	0.5	-
Período de caída de la tasa de aprendizaje	200	-

Las redes neuronales MLP y LSTM representan dos enfoques muy diferentes para modelar relaciones temporales en los datos. En nuestras pruebas para la investigación, el modelo MLP resultó ser más rápido en términos de tiempo de entrenamiento. Esto se debe a que son simples en comparación con las LSTM, ya que no tienen la complejidad adicional de las dependencias a largo plazo en los datos, que es lo que hace que las LSTM sean más costosas computacionalmente. A pesar de su mayor rapidez, el modelo MLP logró superar a la LSTM en términos de rendimiento, lo que indica que este tipo de red es capaz de capturar las relaciones subyacentes en el conjunto de datos de manera más efectiva.

Por otro lado, aunque el modelo LSTM tomó más tiempo para entrenar, no fue capaz de superar el rendimiento de la MLP en nuestras pruebas. Sin embargo, vale la pena destacar que las redes LSTM son buenas en el manejo de secuencias temporales y datos con dependencias a largo plazo, lo cual es una ventaja clave en ciertos dominios de aplicación, como el procesamiento del lenguaje natural y el análisis de series de tiempo. En nuestro caso, a pesar de que las LSTM son técnicamente más avanzadas y flexibles, no cumplieron con nuestras expectativas. Estos resultados subrayan la importancia de la elección del modelo correcto según la naturaleza y la complejidad del problema a tratar.

En la figura 26, se observa el desempeño del modelo MLP al predecir los datos de prueba, comparado con los resultados reales, las predicciones realizadas por nuestro modelo se representan en azul, mientras que los resultados reales se representan en rojo.

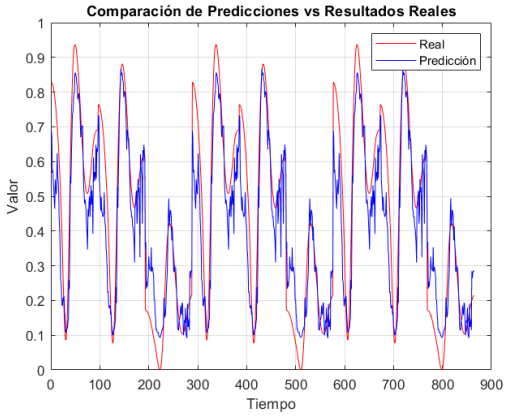

Figura 26: Resultados de comparación de predicciones vs resultados reales.

El trazado de las predicciones se alinea en gran medida con el trazado de los resultados reales, lo que indica que nuestro modelo es capaz de capturar la tendencia general de los datos. Sin embargo, existen ciertas áreas donde el modelo no logra predecir los resultados exactos, lo cual se puede esperar en cualquier problema de aprendizaje automático debido a la naturaleza probabilística y la presencia de ruido en los datos.

La medida cuantitativa del rendimiento del modelo es el error cuadrático medio, en nuestro caso, el MSE es de 0.012119, es un valor relativamente pequeño, lo que indica que las predicciones del modelo son cercanas a los resultados reales. Sin embargo, siempre es recomendable comparar este valor con una métrica de referencia o con los resultados obtenidos por otros modelos para tener un mejor contexto de qué tan bueno es este MSE.

La figura 27 compara las predicciones generadas por la red LSTM con los resultados reales, el eje Y se representa el valor de la variable objetivo, mientras que en el eje X se muestra el tiempo. Las líneas azules representan las predicciones realizadas y las líneas rojas los datos reales, lo permite visualizar la precisión de las predicciones y evaluar si es capaz de capturar los patrones subyacentes en los datos.

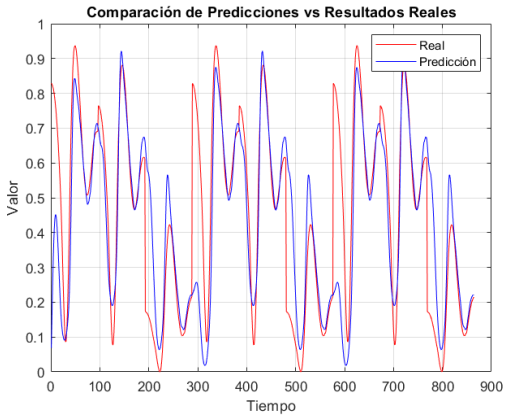

Figura 27: Resultados de comparación de predicciones vs resultados reales.

La red realiza un buen trabajo al predecir los picos más altos en los datos, ya que las predicciones se mantienen dentro del rango esperado, es decir, la red es capaz de capturar y predecir con precisión los cambios más significativos. Sin embargo, se observa cierta discrepancia en la predicción de los picos medios, donde las predicciones no coinciden exactamente con los valores reales, debido a la naturaleza compleja de los datos en estos puntos. El error cuadrático medio es 0.031455, que es una medida cuantitativa del error de nuestras predicciones. Aunque este valor no es alto, indica que hay cierto grado de error en nuestras predicciones. Si hablamos de rendimiento del modelo, un MSE bajo sería lo más deseable, ya que indica una menor discrepancia entre las predicciones y los valores reales.

A pesar de las diferencias en su configuración y la naturaleza de las tareas para las que son más adecuadas, en ambas redes tanto la MLP como la LSTM han demostrado ser capaces de aprender eficazmente a partir de los datos de entrenamiento y realizar predicciones precisas. Sin embargo, en términos de rendimiento, la MLP ha demostrado producir un error cuadrático medio más bajo, lo que indica una menor discrepancia entre las predicciones del modelo y los valores reales, por lo que se sugiere que la red MLP, es más adecuada para nuestro problema específico y conjunto de datos.

7.2. Comparación de desempeño en lazo cerrado

Se presenta una comparación detallada del desempeño de las redes neuronales MLP y LSTM operando los datos en lazo cerrado. Ambos tipos de redes se utilizan para modelar relaciones temporales en los datos, pero difieren significativamente en su estructura y funcionamiento interno. La tabla 12 muestra los hiperparámetros usados para entrenar ambas redes, lo que permite una comparación de su rendimiento. Aunque ofrecen ventajas únicas, su capacidad para un problema específico depende en gran medida de las características de los datos y de la tarea en cuestión. En nuestra investigación, observamos diferencias notables en el tiempo de entrenamiento y precisión del los modelos.

Tabla 12: Comparación de los hiperparámetros de los modelos de redes neuronales para lazo cerrado.

Modelo	**Capas Ocultas**	**Unidades Ocultas**	**Función de Activación**	**Tasa de Aprendizaje**	**Batch Size**
MLP	4	165	ReLU	0.001 (adam)	256
LSTM	2	50	ReLU	0.001 (adam)	64

En nuestra investigación, implementamos dos tipos de redes neuronales, una MLP y LSTM, ambas configuradas en un diseño de lazo cerrado. A pesar de que su finalidad era modelar la secuencia de datos de entrada, estos modelos difieren en términos de su estructura y configuración de hiperparámetros. El modelo MLP consiste en 4 capas ocultas y utiliza 165 unidades ocultas, con una función de activación de ReLU. En cambio, el modelo LSTM contiene 2 capas ocultas con 50 unidades ocultas cada una, también utilizando una función de activación ReLU. Ambos modelos

emplean un optimizador Adam con una tasa de aprendizaje de 0.001, pero difieren en su tamaño de lote o "batch size" durante el entrenamiento. El modelo MLP usa un tamaño de lote de 256, mientras que el modelo LSTM usa un tamaño de lote más pequeño de 64. Estas diferencias en los hiperparámetros reflejan las características únicas de cada tipo de red neuronal y su adaptabilidad para el conjunto de datos usado.

Además de estas diferencias estructurales y de configuración, también observamos variaciones en el tiempo de simulación. La red LSTM se demora más tiempo debido a su complejidad inherente y a su capacidad para manejar dependencias a largo plazo, mientras que la red MLP, con su estructura más sencilla, requiere menos tiempo. Sin embargo, a pesar de esta mayor demanda de tiempo, en nuestra investigación, la red MLP proporcionó mejores resultados en términos de rendimiento en la tarea de modelado de la secuencia.

Las diferencias en los hiperparámetros y tiempos de simulación reflejan las características únicas de cada tipo de red neuronal y su funcioalidad para el conjunto de datos utilizado.

La figura 28 representa los trazados de las predicciones generadas en nuestro modelo en contraste con los valores reales. Nos permite evaluar visualmente la precisión de nuestras predicciones y cómo estas se alinean con los datos verdaderos.

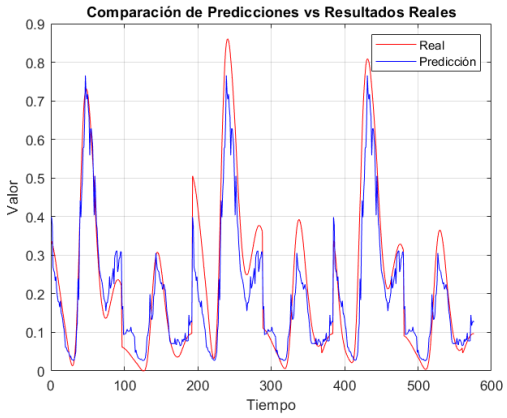

Figura 28: Resultados de comparación de predicciones vs resultados reales.

Analizando la figura 28, se observa que el modelo predice de manera efectiva los picos de los datos reales, sin embargo, en los valores pequeños y medianos, se evidencia una menor precisión, al parecer tiene dificultades para capturar la forma de los datos en estas regiones, lo que resulta una discrepancia entre las predicciones y los datos reales. A pesar de esto, el error cuadrático medio obtenido es 0.0059678. Este valor es bajo, queriendo decir que nuestro modelo no está demasiado alejado de los valores reales. Sin embargo, debemos considerar la posibilidad de optimizar más el modelo, con especial atención a los valores pequeños y medianos, para mejorar aún más este indicador.

La figura 29 presenta una comparación entre las predicciones generadas por nues-

tro modelo LSTM y los resultados reales, se puede observar que nuestro modelo ha logrado capturar la dinámica general de los datos, lo que indica que aprendió patrones importantes en los datos de entrenamiento. Sin embargo, al revisar la gráfica con más detalle, se identifican algunas discrepancias. En los picos altos, por ejemplo, las predicciones tienden a sobrepasar los valores reales. Esto concuerda con lo observado en la figura 25 de progreso de entrenamiento, en la que notamos una posible tendencia al sobre ajuste después de la época 60. Estos resultados nos indican que se debe continuar afinando nuestro modelo y explorando estrategias para evitar el sobre ajuste, como la regularización o el ajuste de los hiperparámetros, con el fin de mejorar la precisión de las predicciones en estos puntos críticos.

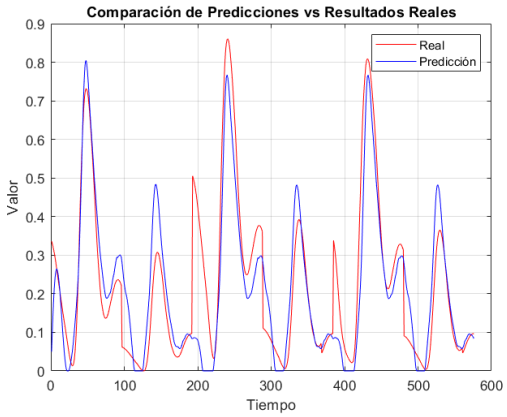

Figura 29: Resultados de comparación de predicciones vs resultados reales.

Los modelos de redes neuronales LSTM y MLP aplicados a nuestro conjunto de datos, han demostrado un comportamiento superior a lo esperado, la principal métrica utilizada para evaluar ambos modelos fue el error cuadrático medio, en la cual la red MLP arrojó un error más bajo en comparación a la LSTM, indicando una precisión en los resultados más alta.

8. Conclusiones y Recomendaciones

Las redes MLP y LSTM han demostrado ser capaces de aprender de manera efectiva y realizar predicciones precisas en los conjuntos de datos tanto en lazo abierto como en lazo cerrado. Sin embargo, en términos de rendimiento, la red MLP ha mostrado un error cuadrático medio (MSE) más bajo en comparación con la red LSTM en ambos escenarios. Esto indica que la red MLP ha logrado una mayor precisión en sus predicciones, capturando de manera más precisa los patrones subyacentes en los datos.

En el entrenamiento de ambas redes, se observa una tendencia al sobreajuste a medida que avanzan las iteraciones. En el caso de la red LSTM en lazo cerrado, se observa un aumento en la pérdida de validación después de la época 60, lo que indica una posible pérdida de capacidad de generalización del modelo. Esto resalta la

importancia de implementar estrategias como la regularización o la optimización de hiperparámetros para mitigar el sobreajuste y mejorar la capacidad de generalización del modelo.

Las redes MLP y LSTM han sido capaces de capturar la tendencia general de los datos y predecir los picos más altos en los resultados. Sin embargo, en los valores pequeños y medianos, se ha observado una menor precisión en las predicciones, especialmente en la red LSTM. Esto sugiere que hay margen para mejorar la capacidad de los modelos para capturar patrones en estas regiones de los datos.

La arquitectura de la red, incluyendo el número de capas ocultas, unidades ocultas y la selección de la función de activación, juegan un papel crucial en el rendimiento del modelo. La configuración adecuada de estos hiperparámetros es fundamental para lograr un buen desempeño. En ambos escenarios, se han utilizado diferentes configuraciones de hiperparámetros para la red MLP y LSTM, lo que destaca la importancia de adaptar la arquitectura de la red a las características específicas del problema y los datos.

A pesar de los buenos resultados obtenidos, existen oportunidades para mejorar aún más el rendimiento de los modelos. Estas mejoras podrían incluir la optimización de los hiperparámetros, la exploración de técnicas de regularización para evitar el sobreajuste, el análisis y la corrección de las discrepancias en las predicciones en los valores pequeños y medianos, así como la exploración de arquitecturas de red más complejas o enfoques de ensamblaje para mejorar la precisión y la capacidad de generalización del modelo.

Referencias

[1] Jens Alex, Lorenzo Benedetti, JB Copp, KV Gernaey, Ulf Jeppsson, Ingmar Nopens, MN Pons, Leiv Rieger, Christian Rosen, JP Steyer, et al. Benchmark simulation model no. 1 (bsm1). *Report by the IWA Taskgroup on benchmarking of control strategies for WWTPs*, 1, 2008. [*Citado en págs.* v *y 27.*]

[2] Carlos Alberto Coelho Belchior, Rui Alexandre Matos Araújo, and Jorge Afonso Cardoso Landeck. Dissolved oxygen control of the activated sludge wastewater treatment process using stable adaptive fuzzy control. *Computers and Chemical Engineering*, 37:152–162, 2012. [*Citado en págs.* v, *21, 22 y 23.*]

[3] Kangyang Chen, Hexia Chen, Chuanlong Zhou, Yichao Huang, Xiangyang Qi, Ruqin Shen, Fengrui Liu, Min Zuo, Xinyi Zou, Jinfeng Wang, et al. Comparative analysis of surface water quality prediction performance and identification of key water parameters using different machine learning models based on big data. *Water research*, 171:115454, 2020. [*Citado en pág. 5.*]

[4] Jorge Dagnino. Coeficiente de correlación lineal de pearson. *Chil Anest*, 43(1):150–153, 2014. [*Citado en pág. 25.*]

[5] Abhirup Dikshit, Biswajeet Pradhan, and M. Santosh. Artificial neural networks in drought prediction in the 21st century–a scientometric analysis. *Applied Soft Computing*, 114:108080, 2022. [*Citado en pág. 7.*]

[6] J Flores, B Arcay, and J Arias. An intelligent system for distributed control of an anaerobic wastewater treatment process. *Engineering Applications of Artificial Intelligence*, 13(4):485–494, 2000. [*Citado en págs. 3 y 4.*]

[7] M Francisco and P Vega. Automatic tuning of model predictive controllers based on multiobjective optimization. *Latin American applied research*, 40(3):255–265, 2010. [*Citado en pág. 6.*]

[8] Luis Octavio González-Salcedo and José Benjamín García-Nuñez. Elaboración de un modelo neuronal artificial para la estimación de turbiedad y proposición de dosificaciones en el tratamiento de aguas residuales de la industria avícola. *Informador Técnico*, 84(1):3–17, 2020. [*Citado en pág. 3.*]

[9] Maged M Hamed, Mona G Khalafallah, and Ezzat A Hassanien. Prediction of wastewater treatment plant performance using artificial neural networks. *Environmental Modelling and Software*, 19(10):919–928, 2004. [*Citado en págs. 1 y 2.*]

[10] Ulf Jeppsson, Christian Rosén, Jessie Alex, John Copp, Krist V Gernaey, M-N Pons, and Peter A Vanrolleghem. Towards a benchmark simulation model for plant-wide control strategy performance evaluation of wwtps. *Water Science and Technology*, 53(1):287–295, 2006. [*Citado en pág. 6.*]

[11] Yiqi Jiang, Chaolin Li, Lu Sun, Dong Guo, Yituo Zhang, and Wenhui Wang. A deep learning algorithm for multi-source data fusion to predict water quality of urban sewer networks. *Journal of Cleaner Production*, 318:128533, 2021. [*Citado en pág. 5.*]

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. [*Citado en pág. 4.*]

[13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. [*Citado en pág. 4.*]

[14] Fei Li, Zhong Su, and Gonging Wang. An effective dynamic immune optimization control for the wastewater treatment process. *Environmental Science and Pollution Research*, pages 1–16, 2021. [*Citado en pág. 3.*]

[15] Wei Lin, Yu Hanyue, and Li Bin. Prediction of wastewater treatment system based on deep learning. *Frontiers in Ecology and Evolution*, 10:1064555, 2022. [*Citado en pág. 4.*]

[16] Jingxian Liu, Zulin Wang, and Mai Xu. Deepmtt: A deep learning maneuvering target-tracking algorithm based on bidirectional lstm network. *Information Fusion*, 53:289–304, 2020. [*Citado en pág. 5.*]

[17] Pedro López Miguel et al. Detección de actividades anómalas en espacios públicos mediante redes neuronales profundas. Master's thesis, Universidad de Alcalà. Escuela Politècnica Superior, 2019. [*Citado en págs.* v, *13, 17, 19 y 20.*]

[18] JM Maciejowski. Predictive control. *Constraints*, 2001. [*Citado en pág. 6.*]

[19] Anthony Njuguna Matheri, Belaid Mohamed, Freeman Ntuli, Esther Nabadda, and Jane Catherine Ngila. Sustainable circularity and intelligent data-driven operations and control of the wastewater treatment plant. *Physics and Chemistry of the Earth, Parts A/B/C*, 126:103152, 2022. [*Citado en pág. 3.*]

[20] Anthony Njuguna Matheri, Freeman Ntuli, Jane Catherine Ngila, Tumisang Seodigeng, and Caliphs Zvinowanda. Performance prediction of trace metals and cod in wastewater treatment using artificial neural network. *Computers & Chemical Engineering*, 149:107308, 2021. [*Citado en pág. 3.*]

[21] Manfred Morari and Jay H Lee. Model predictive control: past, present and future. *Computers & chemical engineering*, 23(4-5):667–682, 1999. [*Citado en pág. 6.*]

[22] Jesús Pérez Guerrero. Redes recurrentes. Master's thesis, Universidad de Sevilla, 2020. [*Citado en págs.* v *y 14.*]

[23] Ivan Pisa, Ignacio Santín, José Lopez Vicario, Antoni Morell, and Ramon Vilanova. A recurrent neural network for wastewater treatment plant effuents' prediction. *Actas de las XXXIX Jornadas de Automática, Badajoz, 5-7 de Septiembre de 2018*, 2018. [*Citado en págs.* v, v, v, v, v, *6, 7, 8, 11 y 12.*]

[24] E Ramirez. Fundamentos teóricos de lodos activados y aereación extendida. *Recuperado de http://documentacion. ideam. gov. co/openbiblio/bvirtual/018834/MEMORIAS2004/CapituloII/1Fundamentosdelprocesodelodosactivados. pdf (Octubr e, 2017)*, 2004. [*Citado en págs.* v *y 21.*]

[25] Elena Martínez Rodríguez. Errores frecuentes en la interpretación del coeficiente de determinación lineal. *Anuario jurídico y económico escurialense*, 1(38):315–331, 2005. [*Citado en pág. 28.*]

[26] Rabee Rustum. *Modelling activated sludge wastewater treatment plants using artificial intelligence techniques (fuzzy logic and neural networks)*. PhD thesis, Heriot-Watt University, 2009. [*Citado en pág. 3.*]

[27] Reinaldo Alberto Sánchez Turcios. t-student: Usos y abusos. *Revista mexicana de cardiología*, 26(1):59–61, 2015. [*Citado en pág. 30.*]

[28] Ignacio Santın, Carles Pedret, and Ramon Vilanova. Mpc tuning in a hierarchical control structure for a wastewater treatment process. *Revista Iberoamericana de Automática e Informática Industrial*, 2014. [*Citado en pág. 24.*]

[29] Mario Francisco Sutil. *Diseño simultáneo de procesos y sistemas de control predictivo mediante índices de controlabilidad basados en normas*. PhD thesis, Universidad de Salamanca, 2011. [*Citado en págs. 20 y 24.*]

[30] Allaka Tarun and Ginuga Prabhaker Reddy. Artificial neural networks for waste-water treatment plant control. In *Proceedings of International Conference on Industrial Instrumentation and Control: ICI2C 2021*, pages 409–419. Springer, 2022. [*Citado en págs. 9, 10 y 11.*]

[31] Jean-David Therrien, Niels Nicolaï, and Peter A Vanrolleghem. A critical review of the data pipeline: how wastewater system operation flows from data to intelligence. *Water Science and Technology*, 82(12):2613–2634, 2020. [*Citado en pág. 3.*]

[32] Amit Kumar Tyagi and Ajith Abraham. *Recurrent Neural Networks: Concepts and Applications*. CRC Press, 2022. [*Citado en págs. 8, 9, 10 y 11.*]

[33] Albert Vilagran Solsona. Facial expression detection using convolutional neural networks. B.S. thesis, Universitat Politècnica de Catalunya, 2018. [*Citado en pág. 15.*]

[34] Ramon Vilanova, Ignacio Santín, and Carles Pedret. Control y operación de estaciones depuradoras de aguas residuales: Modelado y simulación. *Revista iberoamericana de automática e informática industrial*, 14(3):217–233, 2017. [*Citado en pág. 27.*]