Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN CIENCIA E ENXEÑARÍA DE DATOS

Predicción de caudales de entrada y salida en presas hidroeléctricas usando diferentes algoritmos de inteligencia artificial

Estudante: Alberto Fernández Sánchez Alberto Fernández Sánchez

Dirección: Juan Ramón Rabuñal Dopico

A Coruña, junio de 2024.

A mi familia que me ha apoyado en este giro profesional

Agradecimientos

Agradezco a mi tutor, a mi familia, al grupo de investigación RNASA-IMEDIR y a los miembros de este proyecto de ciencias mariñas del CITEEC. Este estudio forma parte del programa de Ciencias Marinas (ThinkInAzul) respaldado por el Ministerio de Ciencia e Innovación y la Xunta de Galicia, con financiamiento de la Unión Europea a través del NextGenerationEU (PRTR-C17.I1) y el Fondo Europeo Marítimo y de Pesca.

Resumen

La predicción ajustada del caudal de entrada en presas desempeña un papel crucial en la gestión de recursos hídricos y la mitigación de riesgos. Este estudio se centra en la presa de Portodemouros (ubicada entre las provincias de A Coruña y Pontevedra), donde se han probado una serie de algoritmos de aprendizaje automático como redes neuronales artificial de Memoria a Corto y Largo Plazo (LSTM), modelos de bagging y boosting (Random Forest y XGBoost) o máquinas de soporte vectorial para predecir el caudal de entrada y salida a la presa. Los resultados demuestran la efectividad bien establecida de estos modelos en la predicción del flujo aplicada a la presa de Portodemouros. Esta comparación ya se ha realizado en otros estudios con modelos matemáticos, programación genética y otros algoritmos de aprendizaje automático. La combinación de datos de precipitación de varias regiones y pronósticos meteorológicos mejora ligeramente la capacidad del modelo para anticipar las variaciones en el caudal de entrada y de salida a la presa. Esta mayor precisión, por pequeña que sea, es esencial para la detección temprana de inundaciones y la toma de decisiones informada en la operación de la presa. En el estudio se concluye que las redes LSTM predicen con un nivel alto de precisión tanto el caudal de entrada como el de salida.

Palabras clave:

Caudal de agua

Modelos de predicción

Lluvia-escorrentía

LSTM

Análisis de datasets

Series temporales

Keywords:

Dam Flow

Prediction models

Rainfall-Runoff

LSTM

Dataset analysis

Time series

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introdución

A lo largo del último siglo se ha intentado predecir el caudal de los ríos a partir de las precipitaciones en las zonas de influencia, los llamados modelos lluvia-escorrentía han proporcionado información muy útil para la gestión de los recursos hídricos y también para la mitigación de los eventos extremos como inundaciones o sequías.

Estos procesos se han abordado, sobre todo en sus inicios, con modelos matemáticos que se han ido perfeccionando a lo largo de las décadas. Muchas de esas técnicas están basadas en el modelo de hidrograma unitario Snyder [3]. Este modelo originalmente presuponía que los sistemas hidrológicos eran lineales e invariantes en el tiempo y estaban limitados únicamente a cuencas hidrográficas calibradas, no eran válidas para cuencas no monitorizadas (eran incapaces de estimar los parámetros de dicha cuenca para poder predecir el caudal). Uno de los aspectos más importantes del trabajo realizado por Snyder [3] fue el establecimiento de las características principales que debía tener el modelo, estas características eran, el área de drenado, la forma, tamaño y distribución de los acuíferos, pendiente del caudal principal, pendiente de las laderas de la cuenca y la acumulación de agua en determinadas regiones debido a obstrucciones del canal. Para ello determinaba una situación base y luego utilizaba la herramienta para poder predecir eventos de lluvia excesiva a partir de la relación entre los parámetros arriba descritos.

Este tipo de modelos se fueron desarrollando con el trabajo de varios grupos de investigadores Chow et al. [4] Rodriguez-Iturbe and Rinaldo [5] Smart [6] los cuales, basados en el modelo inicial, intentaron determinar los parámetros necesarios para definir un algoritmo basado en propiedades geomorfológicas que presuponían la relación entre la geomorfología e hidrología a través del concepto de unidad hidrográfica geomorfológica instantánea (GIUH) y su respuesta directa en la variación de caudal en respuesta al impulso unitario de lluvia excesiva. Resumiendo, estos estudios intentaban mejorar la relación que vincula la forma de la cuenca con su respuesta hidrológica, permitiendo mejorar la predicción de variación de caudal

en dicha cuenca al tener en cuenta de manera más precisa las propiedades geomorfológicas de la misma.

De entre los modelos de hidrograma unitario existen muchas variantes entre las que podemos destacar:

- El modelo original de Snyder.
- El modelo de Taylor y Schwarz (TS) Taylor and Schwarz [7]
- El método de servicio de convervación del terreno (SCS) Soil Conservation Service [8]
- El modelo IUH [9]

Estos modelos previos se han vuelto cada vez más sofisticados ampliando los enfoques lo que ha generado otros muchos como los modelos hidrólogicos globales (GHMs), modelos superficie Terrestres (LSMs) Beck et al. [10] o la generación de algoritmos a partir de información satelital de pluviometría Ciabatta et al. [11].

En las últimas décadas el peso de los modelos de aprendizaje automático en el abordaje de problemas complejos ha aumentado significativamente. La capacidad para integrar y manejar variables muy complejas, extensas, incompletas o difusas con un coste computacional razonable por parte de estos modelos los hace idóneos para este tipo de problemas. Del mismo modo que pasaba con los modelos matemáticos, se han utilizado todo tipo de algoritmos de aprendizaje automático para abordar la tarea de predicción de caudales en función de las precipitaciones. Se ha evaluado del potencial de los modelos hidrológicos derivados mediante programación genética Heřmanovský et al. [12], aplicado programación genética y redes de neuronas artificiales para modelar el efecto de la lluvia en el flujo de escorrentía de una cuenca urbana Rabuñal et al. [13]. También debido al caracter temporal intrínseco en los conjuntos de datos utilizados para crear los modelos se han utilizado redes específicas para capturar ese contexto temporal como son las redes LSTM Dongkyun and Seokkoob [14]. Incluso se utilizaron modelos de Deep Learning además de otros más habituales en este tipo de predicciones de valores contínuos como Regresión Lineal, SVR, Random Forest para regresión o redes convolucionales (TCN) Jo and Jung [15].

A pesar de la gran cantidad de estudios realizados en la predicción de los caudales de entrada de una presa, existen muy pocos estudios acerca del modelaje de la salida de la presa en función del contexto de la misma. Del mismo modo que la lluvia, la temperatura o la humedad del aire influyen en el caudal de entrada de la presa, a la hora de evaluar el caudal de salida hay que tener en cuenta una variable compleja y a veces muy impredecible, el comportamiento humano. Las presas hidroeléctricas son operadas por humanos cuyo criterio viene regido por

unas reglas determinadas por la propietaria de la concesión de dicha presa, pero el operario responsable de la planta tiene cierto margen de maniobra. Es esta diferencia entre lo establecido por las normas y el comportamiento real del operario lo que hace que dicha salida no pueda ser predicha de manera directa.

La mayoría de los estudios se basan en la importancia de conocer el caudal de entrada a los embalses porque en definitiva pueden provocar eventos extremos de alto impacto tanto económico como de vidas humanas. Pero las causas de estos extremos no solo tienen que ver con el caudal de una cuenca en un momento determinado, hay otras variables que tienen tanta importancia o más que la del caudal de entrada, aquellas que confieren el estado actual de la presa o el comportamiento del operario ante dicho estado y un caudal de entrada determinado.

El motivo de este estudio es el de analizar el histórico de datos de operación de una presa para intentar obtener un modelo de predicción de caudal de entrada y salida de dicha presa con el objetivo de conseguir una base para un modelo de ayuda a la toma de decisiones en la gestión de la misma.

Además se pretende no solo verificar la bondad de un modelo de aprendizaje automático (LSTM) para la predicción de los caudales de entrada a una presa del noroeste de España, sino que se pretende establecer la base para obtener otro modelo que sea capaz de tener en cuenta esas mismas variables para poder pronosticar el comportamiento del operario a la hora de generar un caudal de salida a la misma presa, algo que en conjunto puede derivar en eventos extremos de desborde o de sequía.

1.1 Objetivos

El principal objetivo de este proyecto es analizar una serie de registros de operación de una presa hidroeléctrica real para evaluar la aplicación de diferentes algoritmos de aprendizaje automático en la predicción del caudal de entrada y de salida de dicha presa en función de las características más idóneas resultado de dicho análisis. En caso de que dichos algoritmos sean convenientes para la predicción de estas variables supondría una buena base para establecer un sistema de ayuda a la toma de decisiones en la operatividad de este tipo de instalaciones, las cuales por su impacto económico y medioambiental pueden ayudar a la mejora en la seguridad, eficiencia y desarrollo económico de la zona en la que esté establecida.

Capítulo 2
Fundamentos

2.1 Fundamentos Tecnológicos

Para la realización de este proyecto se han utilizado diferentes herramientas software y librerías que se describen a continuación.

2.1.1 Python

La mayor parte del código está desarrollado en el lenguaje de programación python, un lenguaje de alto nivel interpretado y de propósito general. Es conocido por su sintaxis clara y legible, lo que facilita el desarrollo y mantenimiento del código. Python es ampliamente utilizado en diversas áreas, como desarrollo web, análisis de datos e inteligencia artificial lo que lo hace idóneo para los propósitos de este proyecto.

Este lenguaje dispone de una gran cantidad de librerías que permiten aplicar de manera rápida y sencilla funciones y clases complejas de manera fiable y probada. En las siguientes subsecciones se muestran las más importantes utilizadas en este proyecto.

Numpy

NumPy Harris et al. [16], que significa "Numerical Python", es una biblioteca fundamental para la programación en Python, especialmente en el ámbito del cómputo científico y numérico. Esta librería proporciona un conjunto de funciones y herramientas que permiten realizar operaciones matriciales y de álgebra lineal de manera eficiente, lo cual es esencial para muchas aplicaciones científicas y de análisis de datos. En este proyecto se ha utilizado, entre otras tareas, para poder localizar valores no válidos (NaN) en los datos normalizados.

Pandas

Pandas pandas development team [17] es una biblioteca de Python diseñada para el análisis y manipulación de datos. Proporciona estructuras de datos flexibles, como DataFrames, que facilitan la manipulación y limpieza de datos. Pandas es esencial en el ámbito de la ciencia de datos para la preparación y análisis de conjuntos de datos. Para este proyecto el dataframe de pandas es la base para toda la gestión de los datos, desde su filtrado hasta su normalización.

Matplotlib

Matplotlib Hunter [18] es una biblioteca de Python ampliamente utilizada para la creación de gráficos y visualizaciones. Proporciona una variedad de funciones para generar gráficos 2D y 3D, histogramas, dispersión, barras, entre otros tipos de visualizaciones. Matplotlib es una herramienta esencial en el ámbito de la ciencia de datos y el análisis de datos para representar de manera efectiva la información de manera gráfica. En este proyecto se ha utilizado en la mayoría de las gráficas expuestas.

Tensorflow

TensorFlow Abadi et al. [19] es una biblioteca de código abierto para machine learning y deep learning desarrollada por Google. Proporciona una plataforma flexible para la construcción y entrenamiento de modelos de aprendizaje automático. TensorFlow es ampliamente utilizado en la investigación y aplicación de modelos de inteligencia artificial, incluyendo RNA profundas. Para realizar los modelos de este proyecto todos los algoritmos de RNA (como los modelos LSTM) han sido creadas mediante esta librería.

Scikit-learn (sklearn)

Scikit-learn Pedregosa et al. [20] es una biblioteca de aprendizaje automático de código abierto para Python. Ofrece herramientas simples y eficientes para análisis de datos y modelado predictivo, incluyendo algoritmos para clasificación, regresión, clustering y más. Scikit-learn es una opción popular para quienes buscan implementar rápidamente modelos de machine learning en sus proyectos. En este proyecto esta librería se ha utilizado para obtener las métricas de evaluación de los modelos como el error cuadrático medio.

Telebot

Telebot es una biblioteca de Python que facilita la creación de bots para Telegram. Los bots de Telegram son aplicaciones de terceros que pueden realizar diversas tareas, desde responder mensajes hasta proporcionar servicios y realizar acciones automatizadas. Esto permite obtener los resultados de las funciones de manera telemática y es útil cuando se ejecutan scripts

que tardan horas o días en ejecutarse, permiten detectar errores en la ejecución y poder realizar otras taréas mientras el código se está ejecutando. En este caso se ha utilizado para la ejecución de las distintas variantes experimentales de las redes LSTM así como la búsqueda de hiperparámetros de dicha red.

2.1.2 R

R R Core Team [21] es un lenguaje de programación y un entorno de software especializado en estadísticas y análisis de datos. Es ampliamente utilizado en la comunidad estadística y de ciencia de datos para realizar análisis exploratorio, modelado estadístico y visualización de datos. R ofrece una amplia variedad de paquetes y librerías que facilitan tareas específicas en estadísticas y análisis de datos. En este proyecto se ha utilizado el lenguaje para todo el análisis ARIMA.

2.1.3 GitHub

GitHub GitHub [22] es una plataforma de desarrollo colaborativo que utiliza el sistema de control de versiones Git. Permite a los desarrolladores trabajar juntos en proyectos, realizar un seguimiento de las revisiones de código, gestionar problemas y colaborar de manera eficiente. GitHub es esencial para el desarrollo de software en equipo y facilita la contribución de múltiples desarrolladores a un proyecto. Todo el proyecto está en un repositorio de GitHub lo que ha permitido manejar cada una de las versiones del mismo además de permitir trabajar desde varios puestos diferentes.

2.1.4 Entornos de desarrollo integrado (IDE)

En programación, un IDE (Entorno de Desarrollo Integrado, por sus siglas en inglés) es una aplicación que proporciona un conjunto de herramientas y características integradas para facilitar el desarrollo de software. Un IDE es un entorno completo que combina un editor de código, herramientas de compilación, depuración, y a menudo, características adicionales como control de versiones, gestión de proyectos y soporte para diferentes lenguajes de programación. Las principales características de un IDE incluyen:

- Editor de Código: Ofrece funciones avanzadas de edición de texto específicamente diseñadas para la escritura de código, como resaltado de sintaxis, sugerencias automáticas, y formato de código.
- Herramientas de Compilación y Ejecución: Permite compilar y ejecutar programas directamente desde el entorno, facilitando la detección de errores y la visualización de resultados.

- Depurador: Proporciona herramientas para realizar un seguimiento y depurar el código, permitiendo la identificación y corrección de errores durante la ejecución.
- Gestión de Proyectos: Permite organizar y gestionar los archivos de un proyecto, así como acceder fácilmente a recursos externos y bibliotecas.
- Control de Versiones: Algunos IDE incluyen herramientas integradas para trabajar con sistemas de control de versiones como Git, facilitando el seguimiento de cambios en el código.
- Integración con Herramientas Externas: Puede integrar herramientas adicionales como analizadores estáticos, generadores de documentación, y otras utilidades útiles para el desarrollo.

Ejemplos comunes de IDE incluyen IntelliJ IDEA, Eclipse, NetBeans y Visual Studio Code (VSCode).

El uso de un IDE puede mejorar significativamente la productividad de los desarrolladores al ofrecer un entorno centralizado y optimizado para el ciclo de desarrollo de software.

Visual Studio Code

VSCode es un editor de código fuente ligero y potente desarrollado por Microsoft. Es altamente personalizable, admite una amplia variedad de extensiones y proporciona funciones avanzadas de edición y depuración. Su interfaz amigable y su amplia compatibilidad lo convierten en una herramienta popular entre los desarrolladores. VSCode ha sido la herramienta base con la que se han generado todos los archivos con extensión ”py”.

Jupyter

Jupyter es un proyecto de código abierto que desarrolla software interactivo, especialmente en el ámbito de la ciencia de datos y la informática científica. El nombre ”Jupyter” proviene de las combinaciones de los tres principales lenguajes de programación que inicialmente admitía: Julia, Python y R. Una de las aplicaciones más importantes de Jupyter es Jupyter Notebook.

Jupyter Notebook es una aplicación web interactiva que permite la creación y compartición de documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. Es ampliamente utilizado en ciencia de datos y análisis exploratorio de datos debido a su capacidad para integrar código, resultados y explicaciones en un solo documento. Todos los

diferentes experimentos usan funciones creadas en archivos ”py” que luego se importan a los diferentes Notebooks, uno por cada experimento.

2.1.5 Gestión de Archivos

Para el correcto funcionamiento del código es necesario almacenar parte de la información de manera permanente, para ello se utiliza un formato de archivos muy específico diseñado para almacenar datasets como los archivos csv y json.

Comma-Separated Values (CSV)

CSV es un formato de archivo simple que se utiliza para almacenar datos tabulares en forma de texto plano. Cada línea del archivo representa una fila y los valores están separados por comas (u otro delimitador, como punto y coma). Es un formato comúnmente utilizado para intercambiar datos entre diferentes aplicaciones y es fácilmente legible tanto por humanos como por máquinas. Muchos programas y lenguajes de programación, incluidos Python y R, admiten la lectura y escritura de datos en formato CSV. Todos los registros están en formato csv para una gestión más simple de la información.

JavaScript Object Notation

JSON es un formato ligero de intercambio de datos que utiliza una sintaxis legible por humanos. Está basado en un subconjunto de JavaScript, pero se utiliza ampliamente en diversos lenguajes de programación. Los datos en formato JSON están estructurados como pares clave-valor y pueden contener listas y objetos anidados. Se utiliza comúnmente para la transmisión de datos entre servidores y clientes web, así como para el almacenamiento de configuraciones y datos estructurados en general. Python y muchos otros lenguajes proporcionan funciones para la lectura y escritura de datos en formato JSON. En este formato de archivo se guardan todas las métricas de evaluación de cada uno de los experimentos lo que permite una gestión eficiente de los resultados de cada uno de los pasos de cada uno de los experimentos en cada uno de los algoritmos de manera ordenada y fácil de recuperar.

2.2 Fundamentos Teóricos

Para poder analizar en profundidad el problema se han realizado una serie de pruebas preliminares y fijación de parámetros necesarios para poder dar un contexto sólido al problema.

2.2.1 Análisis de dependencia de los datos

Arima

La predicción de caudal en el cauce de los ríos a partir de la precipitación en las zonas de influencia tiene un componente temporal claro. La relación entre las diferencias de caudal y las estaciones del año (que influyen en las precipitaciones) es clara, sobre todo en aquellas latitudes en las que las estaciones están profundamente definidas. A pesar de que la cantidad de parámetros que pueden influir en el cauce de un río son extensos, no es descartable pensar que dichos parámetros puedan converger en una simple fluctuación temporal. Es esto lo que da pie a establecer una única relación como la predominante en dicha predicción, la autoregresiva, o estacionaria. Este tipo de modelizado únicamente tiene en cuenta un único parámetro (la propia salida) dirigida únicamente por la componente temporal de la misma. Esto es, para poder establecer las predicciones de cauces futuros únicamente hace falta generar un modelo con los cauces pasados, ignorando cualquier otro factor externo, dado que se presupone que la información que podría aportar cualquier factor externo ya está intrínsecamente en los valores de salida pasados. Dentro del modelizado de las series temporales existe uno modelo muy extendido llamado modelos ARIMA (AutoRegressive Integrated Moving Average).

La forma general del modelo ARIMA propuesto en la década de los años 70 Box et al. [23] y que parametriza las series temporales desde dos vertientes, el primero es el término autorregresivo (AR) que expone la relación en forma de suma finita entre sucesos de un instante de tiempo y el mismo suceso en períodos anteriores situados a una distancia fija (retardo).

$$y_t = \mu + \sum_{i=1}^{p} \gamma_i y_{t-i} + \varepsilon_t \tag{2.1}$$

donde y_t es el valor actual, μ es una constante, γ_i es el coeficiente de autocorrelación y ε_t es el error.

El segundo es el término de medias móviles (MA) que parametriza las diferencias entre el valor esperado y el observado en un instante de tiempo determinado en función de la suma finita de diferencias entre los valores esperados y observados de instantes de tiempo anteriores

$$y_t = \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{2.2}$$

donde θ_i es el coeficiente de la combinación lineal del error pasado.

Al unir ambos términos en una única ecuación se obtiene la forma general de los modelos

ARMA

$$y_t = \mu + \sum_{i=1}^{p} \gamma_i y_{t-i} + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{2.3}$$

A la hora de definir una serie temporal con un modelo ARMA se debe de fijar el grado tanto de la parte autoregresiva p como el grado del término de medias móviles q. Si la secuencia temporal además presenta una tendencia se puede llegar a modelar incluyendo un nuevo parámetro d que convierta la serie con tendencia en una serie estacionaria mediante la diferencia entre los valores presentes y los retardados a una distancia de d instantes dando lugar al modelado ARIMA.

Correlacion

El coeficiente de correlación fue desarrollado por Karl Pearson en el S.XIX para poder dar una medida estandarizada del grado de asociación lineal existente entre dos variables. El uso de este coeficiente es amplio en todo tipo de campos y ofrece una solución simple y potente a la hora de describir el comportamiento de dos o más variables entre sí. Su cálculo es sencillo y directo dado que únicamente se necesita la varianza de cada una de las variables por separado y la covarianza de ambas.

$$r^2 = \frac{\text{COV}[y, \hat{y}]^2}{\text{VAR}[y] \cdot \text{VAR}[\hat{y}]} \tag{2.4}$$

O puesto de manera más explícita:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \bar{y})^2}{n \sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{2.5}$$

Sus valores van de -1 a 1 siendo estos valores el máximo de correlación negativa y positiva respectivamente, en caso de que el valor del coeficiente de correlación sea 0 se considera que no existe relación lineal entre las variables, es decir, que son incorreladas.

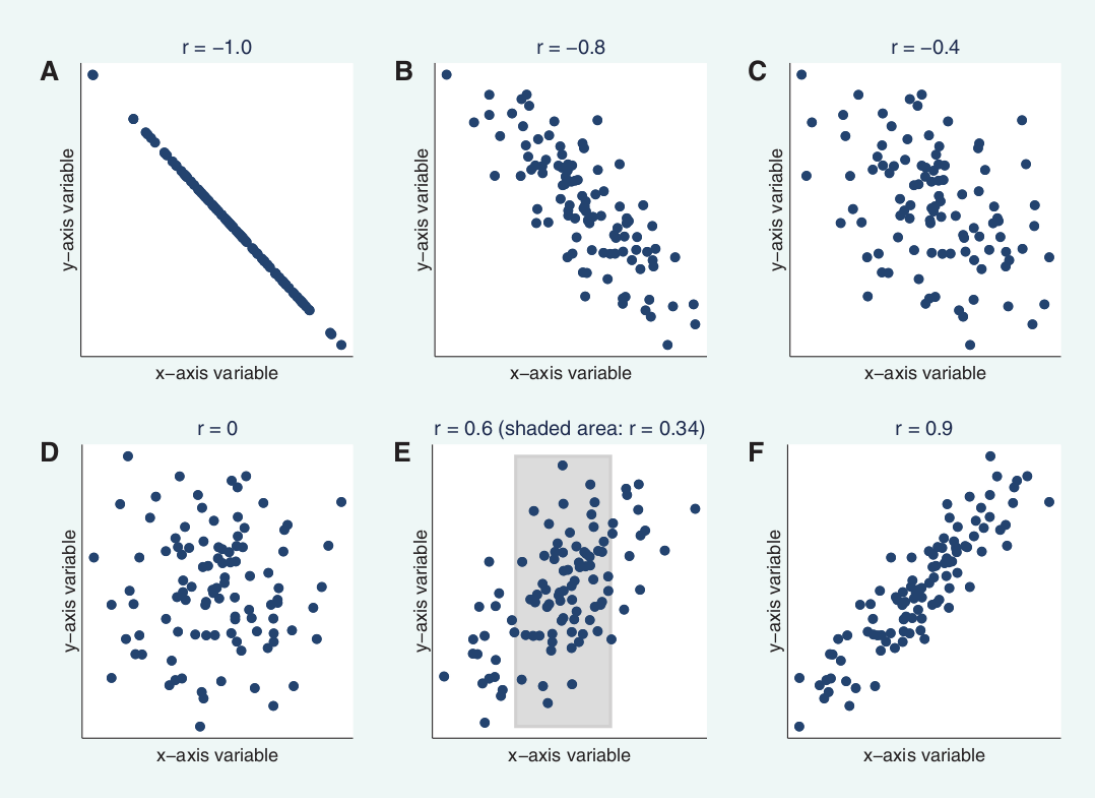

Figura 2.1: A-F, Gráficos de dispersión con datos muestreados a partir de distribuciones normales bivariadas simuladas con coeficientes de correlación de Pearson variables (r). Obsérvese que la dispersión se aproxima a una línea recta a medida que el coeficiente se acerca a -1 o +1, mientras que no hay relación lineal cuando el coeficiente es 0 (D). E muestra con un ejemplo que la correlación depende del rango de los valores evaluados. Mientras que el coeficiente es +0,6 para para toda la gama de datos mostrada en E, sólo es +0,34 cuando se calcula para los datos de la zona sombreada. Fuente: Schober et al. [1]

Análisis de descarga

El análisis de descarga es necesario para poder establecer un contexto en un modelo de lluvia-escorrentía. Lo bueno de conocer los parámetros de descarga del sistema es que permite conocer la relación entre las precipitaciones y el caudal del cauce a determinar integrando variables muy diferentes en un único punto de información. El tiempo que tarda una gota de lluvia caída en un punto del espacio concreto en traducirse en un cambio en el caudal de un río tiene múltiples causas (la distancia física entre la zona exacta de caída de la lluvia y el cauce del río, las características geológicas del terreno, humedad, temperatura...), toda esa información se puede caracterizar de manera general calculando dichos tiempos, es por ello que es un método ampliamente utilizado en los modelos lluvia escorrentía.

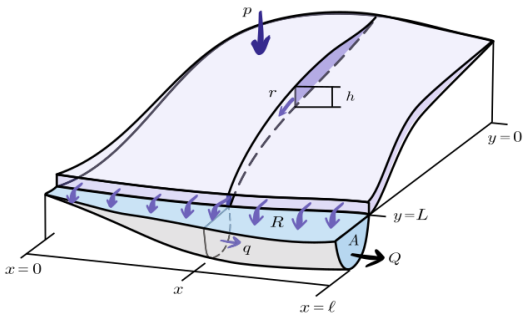

Figura 2.2: Representación esquemática del cauce y de las laderas, mostrando todas las variables que intervienen en la formulación del modelo estocástico lluvia-escorrentía, Fuente: Vallejo-Bernal et al. [2]

En estos análisis lo que se intenta predecir es la cantidad de caudal (o aumento del mismo) que va a llegar al cauce del río a partir de la cantidad de agua precipitada. Como este apartado es un análisis preliminar de contexto de los datos no se intenta calcular todos los parámetros de un análisis de descarga completo, sino únicamente el tiempo que tarda el agua en recorrer el camino de manera conjunta. Es por ello que en lugar de generar un modelo lluvia-escorrentía completo se hace un análisis sencillo calculando el tiempo que tarda desde que hay una precipitación hasta la máxima correlación con una variación en el caudal del río. Únicamente se le añadirá un parámetro adicional a este análisis preliminar, la humedad del terreno estimada mediante la acumulación de precipitaciones pasadas.

Un parámetro influyente en el tiempo de descarga es la saturación de agua del terreno Schnabel [24], es por ello que se calculó la humedad relativa en función del acumulado de lluvia de 1, 2, 3 y 4 semanas previas a la precipitación.

Una vez obtenido esos acumulados, como se intentaba predecir un estado de humedad promedio entre las cuatro áreas de precipitación, debido a que se quiere contabilizar el tiempo transcurrido hasta que el agua llega a la entrada de la presa, lo que se hizo fue caracterizar de manera discreta los acumulados de lluvia en cinco estados diferenciados (seco, semiseco, húmedo, muy húmedo y saturado). Como estos cinco estados no son absolutos, para poder asignar la suma de una cantidad determinada de precipitaciones a cada uno de esos cinco estados lo que se hizo fue asignar cada cantidad a uno de los estados mediante un algoritmo de k-means. Posteriormente se asignó la etiqueta usando el promedio de agua del centroide de cada cluster, esto permite agrupar e identificar por cantidad de agua acumulada de manera rápida y eficiente.

Con los 5 grupos creados se busca la máxima correlación entre la variación de lluvia y variación de caudal calculándolos únicamente en una ventana temporal de 30 días. Así se selecciona, para cada tipo de terreno el máximo de correlación entre el caudal de entrada de la presa y la suma de precipitaciones de los treinta días previos a dicho caudal (usando un coeficiente de correlación de pearson). Esto permite conocer con gran probabilidad cuanto tarda aproximadamente la lluvia en traducirse a caudal efectivo.

2.2.2 Selección de Modelos

Este estudio está centrado en obtener un modelo de inteligencia artificial que tenga en cuenta el contexto no solo espacial (zonas de precipitación) sino también el temporal. Para abordar ambos aspectos se ha decidido por una red LSTM dado que no solo tiene en cuenta la entrada de las variables contiguas en el tiempo sino que también conserva parte del contexto de una ventana temporal de registros pasados.

Debido al análisis de descarga, la capacidad de la red para poder ofrecer predicciones previsiblemente se ve mermada dado que la granularidad de los datos no es lo sufientemente fina como para poder integrar la información temporal en todas sus dimensiones. De todos modos sí que tiene la capacidad para incorporar contextos a más largo plazo debido a su naturaleza. Es por ello que se ha optado por aplicar otros métodos de aprendizaje automático y así realizar un estudio comparado y que apoyen esa posible pérdida de precisión. Los algoritmos seleccionados son:

- Redes de neuronas artificiales completamente conectadas (RNA)
- Random Forest para regresión (RF)
- Máquinas de soporte vectorial para regresión (SVR)
- XGBoost (XGB)

Además, para poder establecer un contexto base se comparará todo con un modelo referencia (denominado Naive) el cual consiste en predecir el valor futuro copiando el último valor conocido. Esto se realiza para tener un modelo básico de referencia sobre el cual poder establecer un contexto. Este tipo de modelos simples se han utilizado en muchos de los estudios aquí mencionados anteriormente Thiesen et al. [25].

RNA

Una red de neuronas artificiales completamente conectada es un modelo de aprendizaje automático inspirado en la estructura y funcionamiento del cerebro humano. Está compuesta

por capas de nodos o neuronas interconectadas, organizadas en al menos tres capas: una capa de entrada, una o más capas ocultas y una capa de salida.

En una RNA, cada conexión entre neuronas tiene un peso asociado que determina la importancia relativa de la señal transmitida. Durante el proceso de entrenamiento, la red ajusta estos pesos para minimizar la diferencia entre las salidas predichas y las salidas deseadas, utilizando un conjunto de datos de entrenamiento.

La capa de entrada recibe las señales del entorno o datos de entrada, las capas ocultas realizan transformaciones no lineales, y la capa de salida produce la predicción o clasificación final. La capacidad de las RNA para aprender patrones complejos y no lineales las hace eficaces en tareas como reconocimiento de patrones, clasificación, regresión y procesamiento de lenguaje natural.

El término ”densa” se refiere a la conectividad completa entre las neuronas de capas adyacentes, lo que significa que cada neurona de una capa está conectada a todas las neuronas de la capa siguiente. Este diseño favorece la captura de relaciones complejas en los datos, permitiendo a las RNA aprender representaciones abstractas y realizar tareas sofisticadas.

LSTM

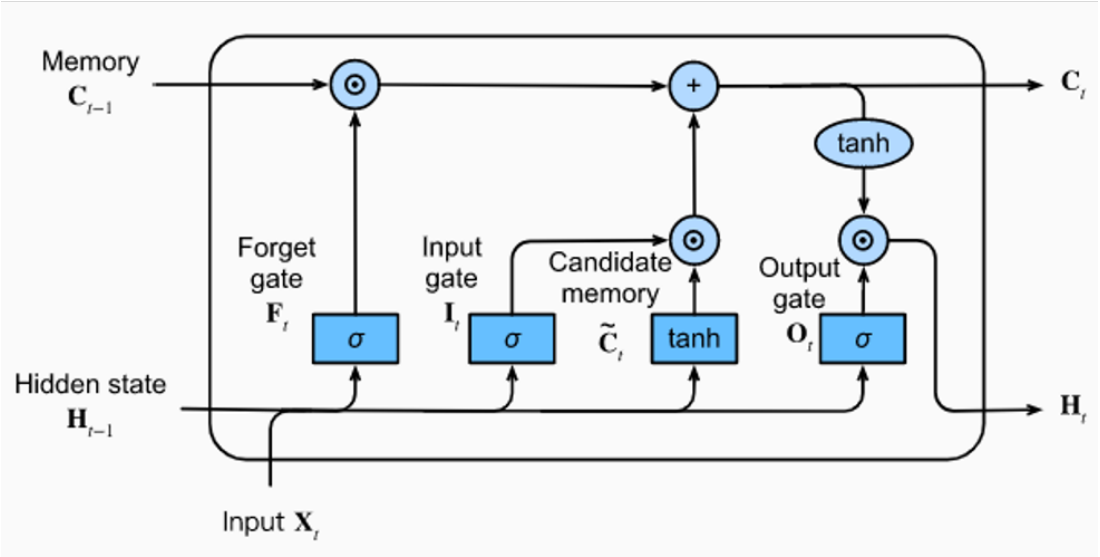

Figura 2.3: Esquema de una neurona de una red LSTM, fuente: https://d2l.ai/chapter_recurrent-modern/lstm.html

Las redes neuronales (RNA) de memoria a largo plazo (LSTM, por sus siglas en inglés) son una variante especializada de las RNA recurrentes (RNN) diseñadas para abordar problemas de modelado de secuencias y series temporales. Las redes LSTM son especialmente adecuadas para capturar dependencias a largo plazo en datos secuenciales y han demostrado ser efectivas en aplicaciones como el procesamiento del lenguaje natural, la predicción climática y, en este caso, la predicción de caudales de entrada en presas.

A diferencia de las RNA tradicionales, que pueden tener dificultades para aprender relaciones a largo plazo debido al problema del desvanecimiento o explosión del gradiente, las redes LSTM incorporan una arquitectura más compleja con celdas de memoria que les permiten retener información durante largos períodos de tiempo. Cada celda LSTM contiene tres puertas principales (ver Figura 2.3): la puerta de entrada, la puerta de olvido y la puerta de salida. Estas puertas controlan el flujo de información y el estado de la celda a través del tiempo.

- Puerta de Entrada (*Input Gate*): Determina qué nueva información se va a agregar al estado de la celda.
- Puerta de Olvido (*Forget Gate*): Decide qué información en el estado de la celda anterior debe ser olvidada o descartada.
- Puerta de Salida (*Output Gate*): Controla la salida del estado de la celda, que se basa en la entrada actual y el estado de la celda anterior.

RF

Random Forest es un poderoso algoritmo de aprendizaje automático utilizado tanto para clasificación como para regresión. En el contexto de la regresión, Random Forest se destaca por su capacidad para modelar relaciones no lineales y capturar patrones complejos en conjuntos de datos.

A diferencia de modelos de regresión lineal, Random Forest utiliza un conjunto de árboles de decisión para realizar predicciones. Cada árbol de decisión se entrena en un subconjunto aleatorio de características y datos de entrenamiento, lo que introduce diversidad en el modelo. Durante la predicción, los resultados de múltiples árboles se promedian para obtener una estimación final.

La fuerza de Random Forest radica en su capacidad para manejar grandes conjuntos de datos, lidiar con características irrelevantes o ruidosas, y mitigar el sobreajuste. Además, al utilizar múltiples árboles, el modelo es más robusto y generaliza bien a datos no vistos.

La salida de Random Forest para regresión es un valor continuo que representa la predicción del modelo para una determinada entrada. Este enfoque permite modelar relaciones complejas y no lineales, lo que lo hace especialmente efectivo en problemas de regresión donde la relación entre las variables no sigue una forma lineal clara.

SVR

Las Máquinas de Soporte Vectorial para Regresión (SVR) son un poderoso enfoque en el campo de la regresión que se basa en los principios fundamentales de las Máquinas de Soporte Vectorial (SVM). A diferencia de los métodos de regresión tradicionales, SVR no busca ajustarse a todos los puntos de datos, sino a aquellos que tienen un impacto significativo en la función de regresión.

La idea central de SVR es encontrar una función que esté lo más cerca posible de la mayoría de los puntos de datos, pero permitiendo cierta flexibilidad al permitir que algunos puntos no se ajusten exactamente. Esto se logra mediante la introducción de una banda o margen alrededor de la función de regresión, donde la mayoría de los puntos de datos deben caer.

La formulación matemática de SVR busca minimizar el error de predicción mientras se controla la amplitud de la banda. La función de regresión resultante está determinada por un conjunto de vectores de soporte, que son los puntos de datos más relevantes para la construcción de la función.

El kernel, una parte fundamental de las SVM, se utiliza en SVR para transformar los datos en un espacio de características de mayor dimensión, permitiendo así la construcción de una función de regresión no lineal. Los diferentes tipos de kernel, como el lineal, polinómico o radial, permiten adaptar SVR a la complejidad de los datos.

SVR es particularmente efectivo en conjuntos de datos no lineales y en situaciones donde la relación entre las variables es compleja. Además, la capacidad de manejar altas dimensiones y la flexibilidad para ajustarse a patrones no lineales hacen que SVR sea una herramienta valiosa en problemas de regresión.

XGB

XGBoost (Extreme Gradient Boosting) para regresión es un algoritmo de aprendizaje automático extremadamente eficiente y preciso que se encuentra entre las técnicas más avanzadas

para la modelización de problemas de regresión. Basado en el concepto de aumento de gradiente, XGBoost combina la potencia de múltiples modelos débiles para formar un modelo fuerte capaz de realizar predicciones precisas.

El algoritmo utiliza árboles de decisión como modelos débiles y realiza el entrenamiento de manera secuencial, corrigiendo los errores cometidos por los modelos anteriores. Cada árbol se ajusta a los residuos del modelo anterior, optimizando así el rendimiento global del modelo.

La fórmula de predicción en XGBoost para regresión se expresa como:

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i)$$

donde:

- $\hat{y}_i$ es la predicción para el i-ésimo ejemplo
- $\phi(x_i)$ es la función de predicción
- K es el número total de árboles
- $f_k(x_i)$ es la predicción del k-ésimo árbol para el i-ésimo ejemplo

XGBoost utiliza una función de pérdida específica para regresión y realiza regularización para evitar el sobreajuste. Además, introduce términos adicionales en la función de pérdida para penalizar la complejidad del modelo, lo que contribuye a su capacidad para manejar datos ruidosos y evitar el sobreajuste.

XGBoost es conocido por su eficiencia computacional y su capacidad para trabajar con conjuntos de datos grandes y complejos. Su rendimiento superior en competiciones de ciencia de datos ha contribuido a su popularidad en la comunidad de aprendizaje automático.

2.2.3 Criterios de evaluación

Para poder medir correctamente la bondad de ajuste de un modelo se utiliza como métrica principal el coeficiente de eficiencia Nash-Sutcliffe debido a su uso extendido en la evaluación de modelos lluvia-escorrentía específicamente Havlíček et al. [26] Ciabatta et al. [11] Heřmanovský et al. [12] Dongkyun and Seokkoob [14] Ansori and Anwar [27] Jo and Jung [15]

Fue propuesto por John R. Nash y James C. Sutcliffe en 1970 y se considera una medida efectiva para evaluar el rendimiento general de un modelo en la simulación de eventos de

escorrentía a lo largo del tiempo. La fórmula básica para el coeficiente de eficiencia de Nash-Sutcliffe se expresa de la siguiente manera:

$$NS = 1 - \frac{\sum_{i=1}^{n}(O_i - P_i)^2}{\sum_{i=1}^{n}(O_i - \bar{O})^2}$$

Donde:

$$\begin{aligned} NS &= \text{Coeficiente de eficiencia de Nash-Sutcliffe} \\ n &= \text{Número de puntos de datos} \\ O_i &= \text{Datos observados en el paso de tiempo } i \\ P_i &= \text{Datos predichos en el paso de tiempo } i \\ \bar{O} &= \text{Media de los datos observados} \end{aligned}$$

El rango de valores de NS (Nash-Sutcliffe) va de $-\infty$ a 1, donde:

- $NS = 1$: La predicción del modelo coincide perfectamente con los datos observados.
- $NS > 0$: El modelo funciona aceptablemente y mejora en comparación con una estimación constante igual al promedio de los datos observados.
- $NS = 0$: El modelo funciona igual que una estimación constante igual al promedio de los datos observados.
- $NS < 0$: El modelo funciona peor que una estimación constante igual al promedio de los datos observados.

Además del valor de NS de cada modelo también se ha utilizado el coeficiente de determinación (R^2) explicado en el apartado anterior. Este coeficiente también ha sido utilizado para evaluar modelos de lluvia-escorrentía Zhong et al. [28] Ansori and Anwar [27] Jo and Jung [15] y que es una medida estadística que proporciona una indicación de la calidad del ajuste de un modelo de regresión a los datos observados. Más específicamente, R^2 se define como la proporción de la varianza total de la variable dependiente que es explicada por el modelo de regresión.

$$R^2 = 1 - \frac{\text{Varianza no explicada por el modelo}}{\text{Varianza total de la variable dependiente}}$$

El valor teórico de R^2 está en el rango de 0[1] a 1. Un R^2 de 1 indica que el modelo explica

[1] En este estudio se ha utilizado la librería $sklearn.metrics.r2\_score$ para calcular R^2 la cual sí permite valores negativos, dichos valores negativos no aportan ningún tipo de información y se interpretan del mismo modo que 0.

perfectamente la variabilidad de la variable dependiente, mientras que un R^2 de 0 indica que el modelo no proporciona ninguna mejora en la predicción en comparación con simplemente utilizar la media de los valores observados.

Capítulo 3

Estado de la cuestión

A la hora de realizar la predicción de caudal de entrada a partir de la lluvia de las zonas de influencia (modelo lluvia-escorrentía) se ha abordado de maneras muy diferentes, tanto modelos matemáticos directos como todo tipo de algoritmos de inteligencia artificial.

3.1 Modelos matemáticos

Desde el punto de vista matemático existen múltiples algoritmos surgidos por los grandes grupos de investigación hidrológica, la mayoría públicos directamente financiados por los gobiernos de diferentes países. Esto hace que existan gran cantidad de modelos, algunos de los cuales se detallan a continuación:

- MOPSO Amirreza et al. [29]
- SCS Vargas-Garay et al. [30] Fan et al. [31]
- TOPMODEL Zhong et al. [28]
- HEC-RAS Costabile et al. [32]
- SM2RAIN Ciabatta et al. [11]
- TMPA Ciabatta et al. [11]
- SWAT A.R et al. [33]
- GR4J Ansori and Anwar [27]

Además de estos modelos específicos también se han aplicado técnicas estadísticas o matemáticas que atacan directamente alguna de sus características principales (componente temporal, relación entre sus variables y la salida, etc.). Entre estas técnicas destacan los análisis

de dependencia temporal ARIMA Zhang et al. [34] Valipour [35] Pooja Verma and Swastika Chakraborty [36] Zhao et al. [37] Dhote et al. [38], el uso del coeficiente de correlación para evaluar modelos en general Waldmann [39] o específicamente para el uso de modelos de lluvia-escorrentía Razmkhah et al. [40].

Por otra parte un aspecto específico de este problema en concreto es el modelizar el tiempo que transcurre desde que una precipitación toca el suelo hasta que se traduce en un cambio de caudal (análisis de descarga), los modelos de análisis de descarga se también han sido un complemento importante en el perfilado de sistemas lluvia-escorrentía Vallejo-Bernal et al. [2] y dentro de los mismos el estado de saturación de agua del terreno es un factor importante que también ha sido modelizado Meier et al. [41]

Otro enfoque del problema es dividir las distintas variables en distribuciones de probabilidad y con ello intentar aproximar la distribución real del sistema usando como función objetivo la minimización de la distancia entre las distribuciones según el criterio Kullback–Leibler (KL divergence) Thiesen et al. [25]

La mayoría intenta predecir el caudal dividiendo un problema complejo (caudal a partir de las condiciones espacio-temporales) en un agregado de problemas más sencillos de influencia directa (caudal a partir de la lluvia, humedad, estado del terreno) o incluso de manera indirecta como sería el estudio de la erosión del terreno A.R et al. [33].

3.2 Modelos basados en aprendizaje automático

Debido al gran uso que estas técnicas están teniendo recientemente han sido muchos autores los que han aplicado este tipo de herramientas a la predicción de caudal a partir de las condiciones meteorológicas pasadas. Del mismo modo que sucede con los modelos matemáticos, se han aplicado una gran variedad de algoritmos de inteligencia artificial, desde modelos de simulación como MonteCarlo Razmkhah et al. [40], modelos más generales como redes de neuronas artificiales Aytek et al. [42] Rabuñal et al. [13], hasta redes convolucionales pasando por regresión lineal, máquinas de soporte vectorial para regresión o random forest para regresión Jo and Jung [15]. Debido a la complejidad del problema los modelos lluvia-escorrentía también han sido susceptibles de ser abordados desde paradigmas de computación evolutiva Heřmanovský et al. [12] Havlíček et al. [26]Aytek et al. [42]Rabuñal et al. [13].

Modelar el comportamiento humano con algoritmos de aprendizaje automático es algo

que se lleva haciendo desde hace algunos años para múltiples propósitos, desde algoritmos para predecir el comportamiento de conductores de vehículos Kolekar et al. [43] hasta el modelado de la interacción de personas con otras personas Gloor [44]. Desde el punto de vista de las tecnologías utilizadas hay estudios con redes LSTM Basavaraj et al. [45], random forest o regresión logística Robila and Robila [46]. Esto demuestra que el uso de algoritmos de aprendizaje automático en el modelado del comportamiento humano es una práctica útil y bien documentada.

En este trabajo se pretende utilizar la versatilidad de los modelos basados en RNA para poder generar un modelo que pueda predecir una variable, el caudal de entrada y salida de una presa hidroeléctrica, que depende de muchos factores a su vez complejos como son el clima, la orografía o el factor humano. Este tipo de objetivos tienen como último fin el poder modelar el comportamiento de una presa hidráulica, esto es importante no solo por el impacto económico que estas instalaciones pueden generar sino también por el impacto ambiental o incluso de seguridad para los habitantes del entorno de dichas instalaciones.

El poder predecir el comportamiento puede permitir mejorar la eficiencia en la generación de energía eléctrica (lo que permite aumentar las ganancias de los operadores de la instalación), también permite reducir la cantidad de recursos necesarios para la generación de energía, adecuando el gasto de esos recursos en los momentos que menor impacto medioambiental produzcan no solo en el presente sino también evitando situaciones extremas en el futuro.

Desde el punto de vista de seguridad es muy importante el poder prever eventos extremos (de sequía o de inundación) debido a que la gravedad de las consecuencias de dichos eventos es máxima. Cuanto más margen de predicción se obtenga más fácil es poner todas las medidas necesarias a tiempo para evitar dichos eventos.

Por todo esto, los proyectos de modelización lluvia-escorrentía son muy numerosos y enfocados desde muy diversos puntos de vista. Este proyecto pretende materializar la aplicación de varias técnicas en una presa real, lo que permite no solo aplicar muchas de las técnicas aprendidas durante el grado, sino que también permite compararlas y demostrar que estas técnicas son perfectamente válidas para este tipo de problemas en un entorno real.

Capítulo 4

Metodologia CRISP-DM

4.1 Metodología Crisp-DM

Para la realización de este proyecto se ha seguido la metodología Crisp-DM la cual se basa en seis procesos que se ejecutan de manera iterativa y circular dando como resultado un desarrollo incremental y basado en una política de mejora contínua Chapman et al. [47]. Dichos procesos se detallan a continuación:

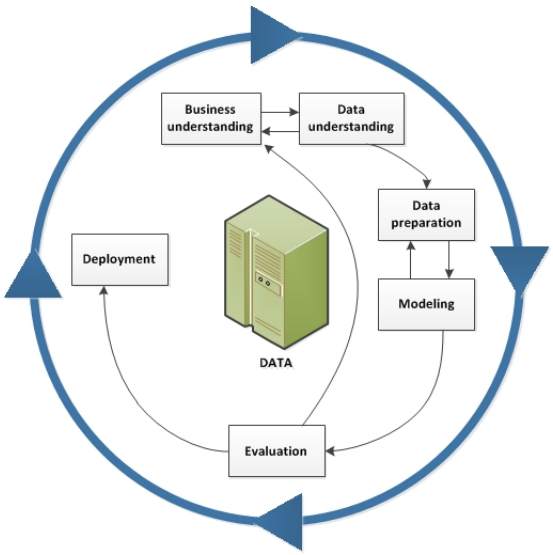

Figura 4.1: Metodología Crisp-dm Fuente:https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview

1. Entendimiento del proceso o negocio: Analizar los aspectos clave del contexto de desarrollo del proyecto, entender sus relaciones, sus particularidades y su entorno.

2. Entendimiento de los datos: Conocer su tipología, sus límites, sus relaciones y su signi-

ficado entre otros.

3. Preparación de los datos: Normalización, tratamiento de datos anómalos o faltantes, acotar los datos al problema en cuestión, integrarlos y agruparlos.

4. Modelado: Diseño, desarrollo y entrenamiento de los diferentes algoritmos que procesarán los datos.

5. Evaluación: Obtención de las métricas que miden la calidad del procesado por parte de los algoritmos y extraer las conclusiones pertinentes entorno a dichas métricas.

6. Despliegue: Poner a disponsición del proceso aquellos algoritmos que han superado la fase de evaluación para integrarlas al proceso.

En este proyecto se ha adaptado el problema a dicha metodología intentando encajar todas las tareas de manera lo más fielmente posible simulando un entorno de negocio real.

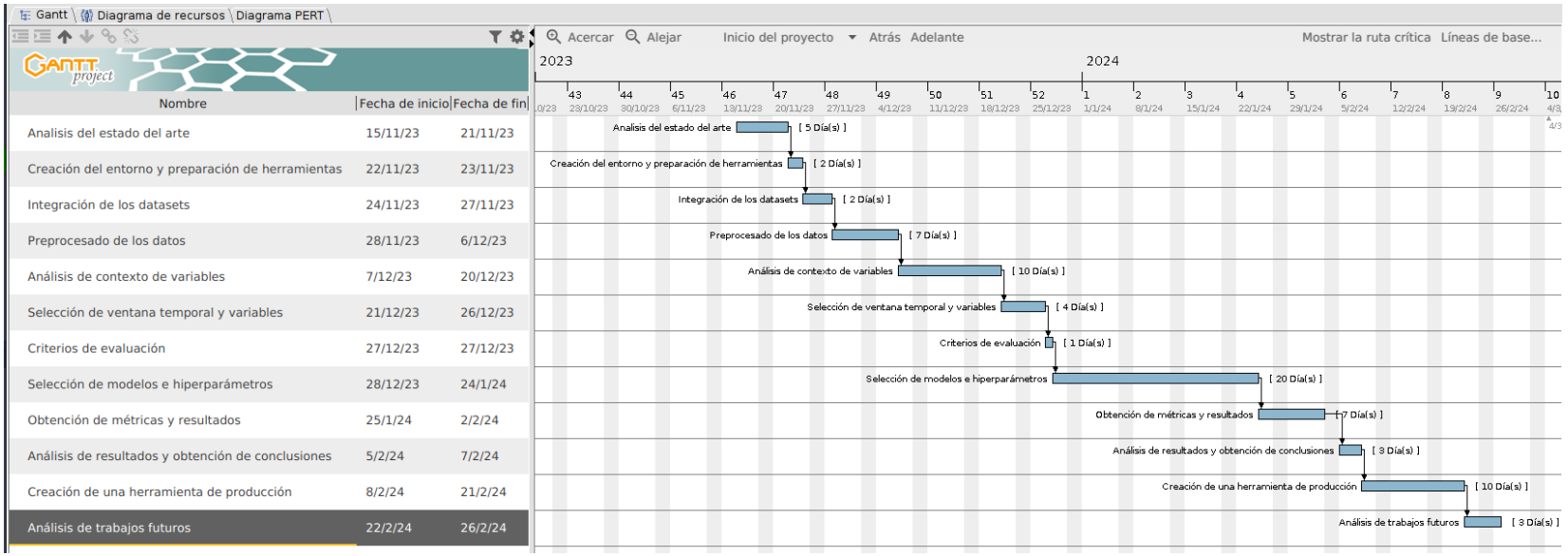

Figura 4.2: Diagrama de Gantt con la relación secuencial de tareas de este proyecto

Como se puede comprobar los pasos 2 (Entendimiento de los datos) y 3 (Preparación de los datos) están muy interrelacionados en este proyecto y esto provoca que se vaya hacia atrás en alguna ocasión. Esto está contemplado en la metodología CRSIP-DM y se considera una práctica válida debido al carácter flexible de esta metodología.

Dichas tareas, su definición y su duración prevista se desarrolla como sigue.

4.1.1 Entendimiento del proceso o negocio: Análisis del Estado del Arte

- Revisar las investigaciones y tecnologías existentes en modelos de lluvia escorrentía, estimación de caudales a partir de variables geológicas y meteorológicas, modelos de aprendizaje automático en la predicción del comportamiento humano.
- Identificar las mejores prácticas y enfoques utilizados por otros en proyectos similares.
- Definir los objetivos y establecer aquellos estudios que serán base para la comparativa de este estudio.
- Duración estimada: 20 horas

4.1.2 Entendimiento de los datos: Análisis previo de los datos

- Estudiar el formato de los datos
- Verificar los rangos de fechas de interés de las diferentes fuentes temporales
- Duración estimada: 5 horas

4.1.3 Preparación de los datos: Creación del Entorno y Preparación de Herramientas

- Configurar el entorno de trabajo, incluyendo librerías y entornos de programación de python.
- Duración estimada: 16 horas

4.1.4 Preparación de los datos: Integración de los Datasets

- Integrar las diferentes fuentes de datos en una sola
- Seleccionar las fechas de los ejemplos
- Realizar limpieza de datos y tratamiento de valores atípicos y valores faltantes

- Duración estimada: 16 horas

4.1.5 Preparación de los datos: Preprocesado de Datos

- Normalizar, estandarizar y transformar los datos según sea necesario.
- Duración estimada: 56 horas

4.1.6 Entendimiento de los datos: Análisis de Contexto de Variables

- Verificar la fidelidad de las reglas de llenado
- Aplicar análisis ARIMA,
- Realizar el análisis de descarga
- Realizar el análisis de correlación.
- Duración estimada: 80 horas

4.1.7 Modelado: Selección de Ventana Temporal y Variables

- Determinar la ventana temporal adecuada para el análisis.
- Seleccionar las variables relevantes para el modelo.
- Duración estimada: 32 horas

4.1.8 Modelado: Criterios de Evaluación

- Definir los criterios de evaluación para medir el rendimiento de los modelos.
- Establecer umbrales de aceptación para las métricas clave.
- Duración estimada: 8 horas

4.1.9 Modelado: Selección de Modelos e Hiperparámetros

- Evaluar diferentes algoritmos de modelado.
- Ajustar hiperparámetros para optimizar el rendimiento del modelo.
- Duración estimada: 160 horas

4.1.10 Evaluación: Obtención de Métricas y Resultados

- Entrenar los modelos seleccionados con los datos de entrenamiento.
- Evaluar el rendimiento utilizando datos de validación y prueba.
- Obtener métricas y resultados para cada modelo.
- Duración estimada: 56 horas

4.1.11 Evaluación: Análisis de Resultados y Obtención de Conclusiones

- Interpretar los resultados y compararlos con los criterios de evaluación establecidos.
- Extraer conclusiones sobre la efectividad de los modelos.
- Duración estimada: 24 horas

4.1.12 Despliegue: Creación de una Herramienta de Producción

- Desarrollar una herramienta basada en los modelos seleccionados para su uso en entornos de producción.
- Duración estimada: 80 horas

4.1.13 Evaluación: Análisis de Trabajos Futuros

- Identificar posibles mejoras y expansiones del proyecto.
- Investigar áreas para futuros desarrollos y perfeccionamientos.
- Duración estimada: 24 horas

4.1.14 Preparación de la memoria

- Duración estimada: 40 horas

4.2 Sistema de gestión de presas hidroeléctricas

4.2.1 Dataset

Para este trabajo se han utilizado los registros correspondientes a una presa localizada en las coordenadas 42°50'47"N 8°08'26"W entre las provincias de A Coruña y Pontevedra Figura 4.3, en el Noroeste de España llamada presa de Portodemouros. Esta presa fue construída en 1964 y forma parte del el cauce del río Ulla.

Tiene una longitud en su radio mayor de 469m y una altura máxima de 93m lo que le confiere una capacidad total de 293hm^3. Dispone de varias compuertas de desague con una capacidad total de 1550 m^3/s, de los cuales una parte va para turbinado siendo capaz de generar un máximo de 95MW de potencia total.

Para poder hacer predicciones de caudal a la entrada y salida de la presa se dispone de 4554 registros diarios desde 2009 hasta 2022 de las principales métricas de la presa (caudal de entrada al final del día, cota de llenado en volumen al final del día, cota de llenado en metros al final del día y caudales de cada una de las compuertas de salida de la presa a lo largo del día por separado). Esto con respecto a los registros de la presa, pero también se dispone tanto de los registros de lluvia de 4 zonas de influencia en el cauce del Rio Ulla aguas arriba: Arzúa, Melide, Olveda y Serradofaro, estos cuatro pluviómetros están operados por Meteogalicia y sus registros son públicos. Además de los datos de registro acumulado diario en cada una de estas zonas, también se disponen de los datos de previsiones meteorológicas para la zona en conjunto a uno, dos y 3 días ofrecidas también por Meteogalicia para todo el rango de fechas necesario para el estudio. Todos estos registros tienen una etiqueta de fecha para permitir su integración en un único dataset.

Figura 4.3: Situación geográfica de la presa y las estaciones meteorológicas

4.2.2 Analisis de contexto de las variables

Verificación de la fidelidad de las reglas de llenado

El operario de la presa trabaja con unas reglas concretas que definen cuál deberá ser su comportamiento en cada situación posible ver figura 4.4

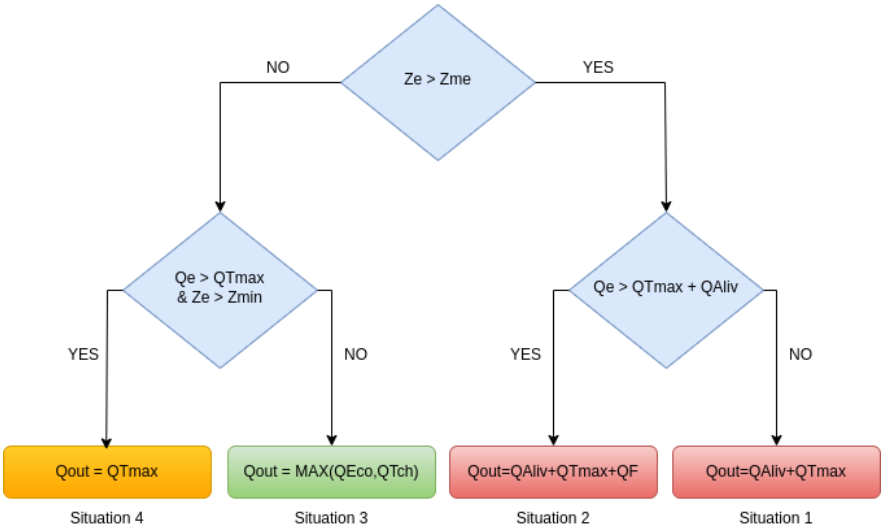

Figura 4.4: Reglas de operación

En estas reglas se le indica al operario cuanto deberá de abrir las diferentes compuertas que conforman el caudal de salida ($Qsalida$) y que consisten en la compuerta de turbinado ($QTch$), la compuerta de aliviadero ($Qaliv$) y la compuerta de fondo ($Qfondo$). Además de estas salidas la presa contiene una cuarta que conforma el caudal ecológico del cauce ($Qeco$), que está regulado por la administración pública (la Xunta de Galicia en este caso) y sobre el cual el operario no tiene ningún control.

Las reglas de operación fijan la apertura de cada una de las puertas en función de diferentes variables como son el nivel de llenado de la presa (Ze) y el caudal de entrada (Qe).

Además de todas estas variables las reglas de operación tienen en cuenta los límites de la presa como el máximo (Zme) y el mínimo ($Zmin$) de llenado de la presa y el caudal máximo de turbinado ($QTmax$).

La combinación del estado de estas variables le proporciona al operario cuatro posibles situaciones en las que puede manejar cada una de las compuertas de salida.

Una vez se dispone del dataset completo se procede a realizar el análisis de las reglas de llenado que tiene el operario, esto se hace como verificación de que no se puede predecir con exactitud el caudal de salida con una simple fórmula matemática derivada de las reglas de llenado que se le proporcionan al operario.

En este análisis lo que se pretende es ver si existen diferencias entre el caudal de salida teórico que debería haber en cada momento en función de las reglas de llenado y los registros de caudal de salida real ver figura 4.5.

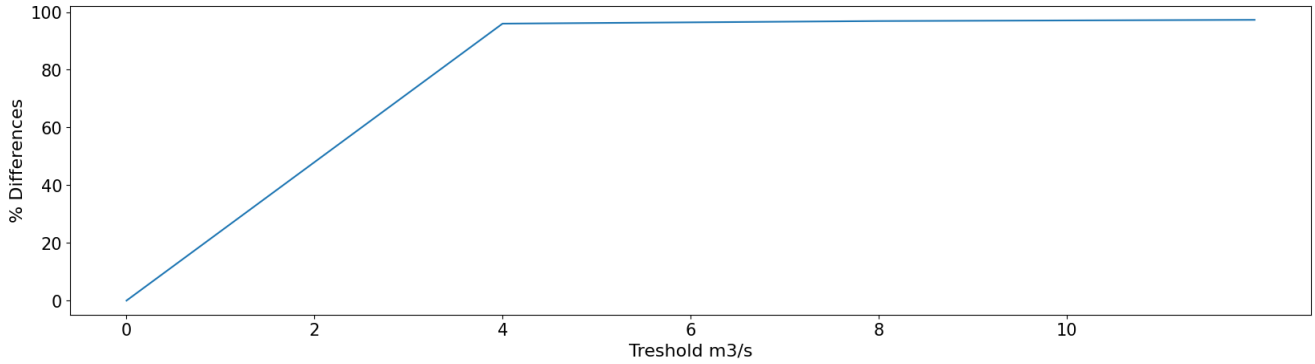

Figura 4.5: Diferencia en el caudal teórico y el registrado en el histórico de operación. El eje de abscisas muestra el margen de diferencia entre el teórico y el real y el eje de ordenadas muestra el porcentaje de registros en el que ambos valores coinciden dentro de ese margen

Cómo se puede comprobar en el grafico 4.5, las reglas se siguen pero no de manera exacta, los registros teóricos y los reales divergen al menos en $2m^3/s$ en casi del 50% de los días.

Si se analiza para cada una de las situaciones fijadas en las reglas de operación se puede verificar que en diferentes situaciones el operario actúa de manera diferente ver Figura 4.6.

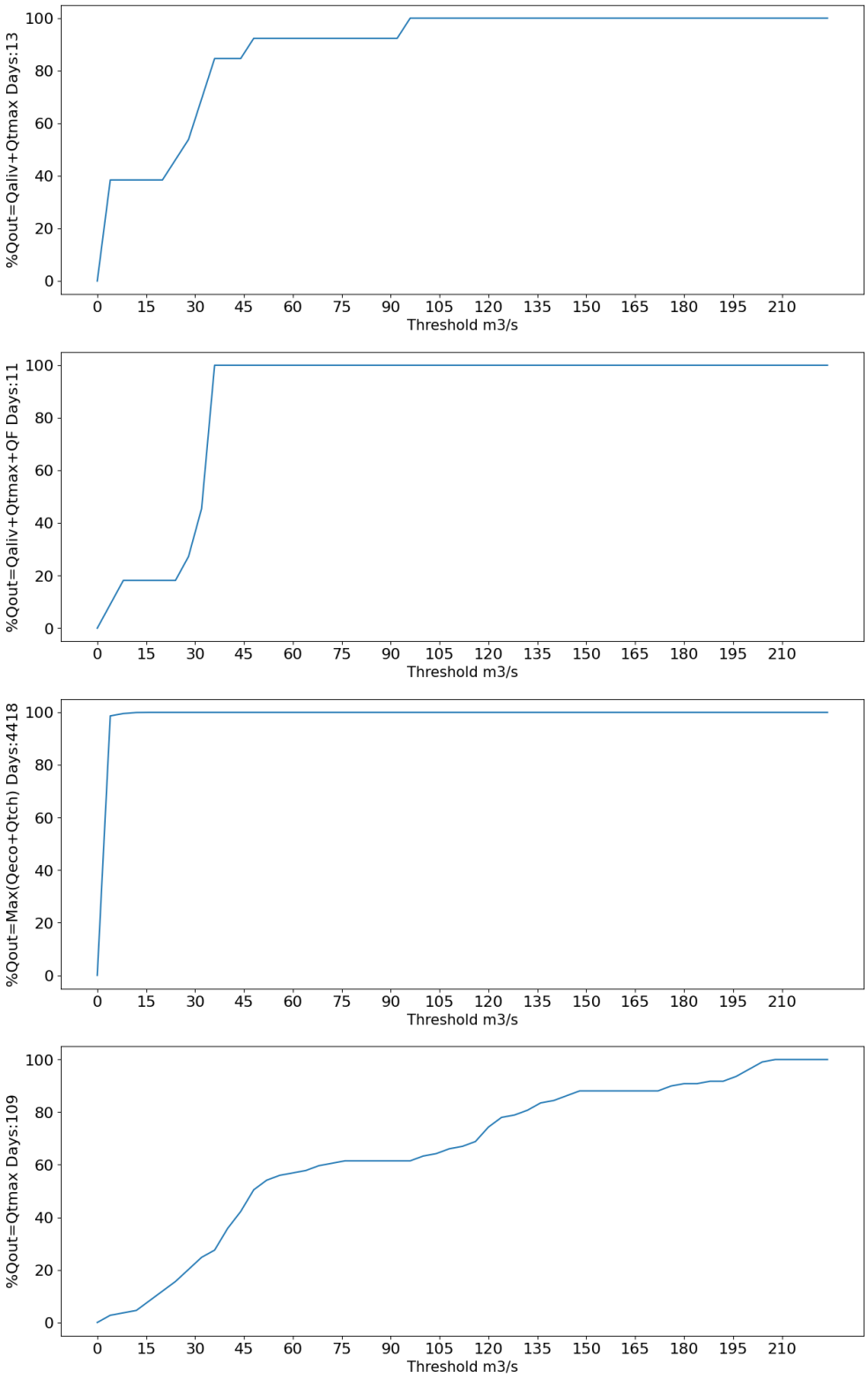

Figura 4.6: Diferencia en el caudal teórico y el registrado en el histórico de operación por cada situación. El eje de abscisas muestra el margen de diferencia entre el teórico y el real y el eje de ordenadas muestra el porcentaje de registros en el que ambos valores coinciden dentro de ese margen

La situación más común es aquella en la que el nivel de la presa no llega al máximo y además el caudal de entrada no supera el máximo de caudal turbinable, denominada situación 3 (4418 días), esta es una situación de operativa normal y se debe de turbinar a discrección siempre y cuando se supere el mínimo de caudal ecológico. Esta situación es la más común a lo largo del año y dicha situación no supone ningún riesgo para las instalaciones ni su entorno.

La situación 4 (109 días) es de transición, el caudal de entrada está por encima del caudal turbinable máximo pero dado que los niveles de la presa no están en el límite no supone un riesgo.

Las situaciones 1 y 2 (que contienen 13 y 11 registros cada una) son las que representan eventos de riesgo, de avenida, la situación 1 por el estado propio de la presa que está al límite aunque el caudal de entrada no sea extremo y la situación 2 por ambas partes, el estado límite de llenado de la presa y el caudal de entrada extremo. Es en estas dos situaciones en las que es necesario la apertura de las compuertas auxiliares (fondo y aliviadero) para reducir el volumen embalsado.

Por lo tanto, las situaciones menos comunes son las más importantes de predecir y como se puede ver en la figura 4.6 su diferencia de caudales con respecto a las reglas de operación divergen más que la situación más común. Además la situación 4 (109 días) es la que puede desembocar en las dos primeras y la que más diferencias de caudal teórico con respecto al real hay.

Por otra parte la situación 3 (4418 días) aunque sí es la situación más común también es susceptible de llevar a la presa a una situación extrema debido a que es la situación que puede predecir un estado de sequía de la presa, por lo tanto el predecir correctamente esta situación también es importante.

Todo eso confirma que las reglas de llenado no son suficientes como para explicar el funcionamiento de la presa y no son un buen predictor de las situaciones que se pretenden describir en este trabajo.

Es por ello pertinente abordar las predicciones del caudal de salida con algún modelo estadístico o de aprendizaje automático más allá de las reglas de operación.

Arima

Para poder trabajar con los datos y puesto que son series temporales primero hay que verificar que el peso de las predicciones no recaen en su totalidad en la propia variable de salida en forma de una serie temporal. Si fuese el caso se podría verificar que los caudales no dependen de ningún factor externo más allá de los propios caudales pasados.

Se ha intentado definir la serie temporal mediante un modelo ARIMA que explica la cantidad de caudal de entrada a la presa en un punto determinado en el tiempo a partir de los valores del mismo caudal de entrada en instantes de tiempo anteriores. Este ajuste únicamente se ha realizado al caudal de entrada debido a que el de salida depende de este, por lo que, si se confirma que no es un modelo únicamente temporal en la entrada, tampoco lo es en la salida.

Utilizando la librería Tseries de R se intentó ajustar un modelo ARIMA fijando la frecuencia a 365 días. Una vez hecho esto se obtuvieron las gráficas de las auto-correlaciones y medias móviles ver figura 4.7.

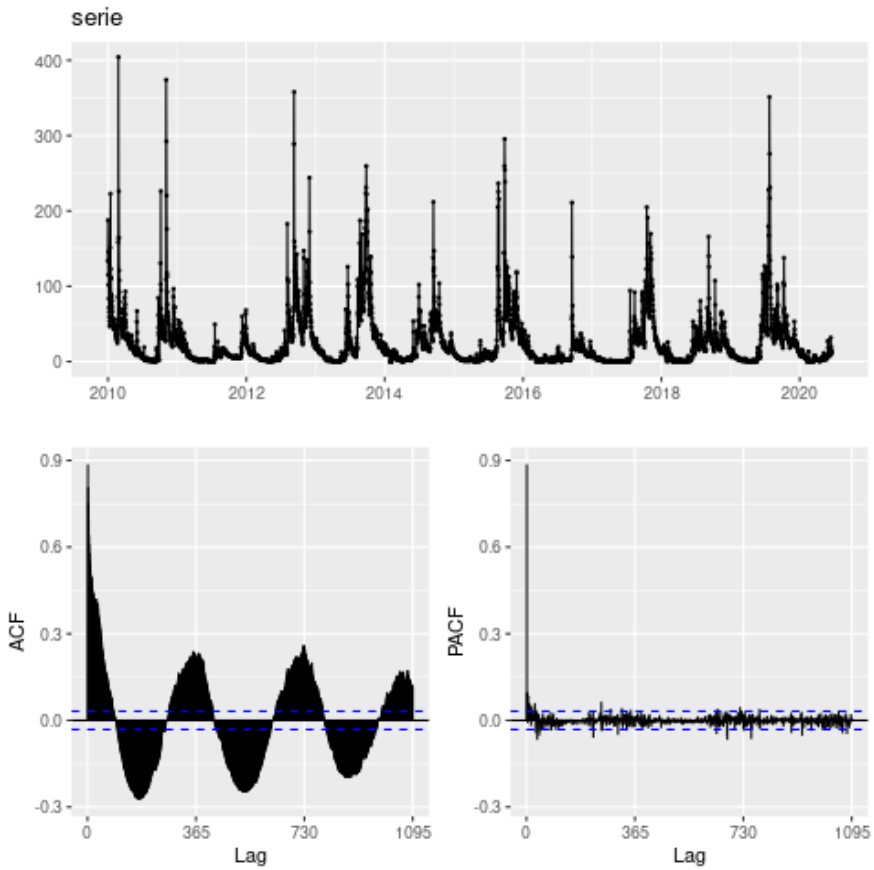

Figura 4.7: Gráficas de registro de la serie (arriba), autocorrelaciones parciales (ACF) y medias móviles (PACF)

En esta gráfica únicamente se puede determinar que no existe tendencia, además se puede intuir que existe una componente temporal. Si se observan las gráficas de autocorrelaciones parciales y medias móviles no se puede ver que exista una única componente (dado que en ninguna de ellas se puede apreciar un pico repetitivo a una distancia fija). Como únicamente con la gráfica no se puede proponer ningún modelo se calcula mediante la función autoarima.

Con dicha función autoarima se obtuvo un modelo autor-regresivo de orden 4, de medias móviles con orden 2 y cuya media no es cero tal y como se puede ver en la figura 4.7.

ARIMA(3,0,2) with non-zero mean: Inf
ARIMA(4,0,2) with non-zero mean: 32356

Best model: ARIMA(4,0,2) with non-zero mean

Series: serie
ARIMA(4,0,2) with non-zero mean

Coefficients:

	ar1	ar2	ar3	ar4	ma1	ma2	mean
	-0.0865	0.1261	0.6105	0.1201	0.8762	0.6040	24.3091
s.e.	0.1289	0.1152	0.0757	0.0234	0.1285	0.0928	2.9015

$\sigma^2 = 277.9$: log likelihood $= -16169.98$
AIC $= 32355.96$ AICc $= 32356$ BIC $= 32405.95$

Figura 4.8: Resultado de la función autoarima

Para seleccionar el modelo utiliza el criterio de información de Akaike (AIC) en su versión corregida (AICc)

$$\text{AIC}(p,q) = -2\log\left(L(\hat{\theta}_{(p,q)})\right) + 2k$$

$$\text{AICc}(p,q) = \text{AIC}(p,q) + \frac{2(k+1)(k+2)}{T-k-2}$$

Donde:

- p es el coeficiente de la componente auto-regresiva
- q es el coeficiente de la componente de medias móviles
- k es el resultado de la suma de p y q más uno si el modelo tiene componente constante
- T es el tamaño total de la serie
- L es la función de máxima verosimilitud

Este criterio mide la calidad del ajuste del modelo a la serie a la vez que penaliza el aumento en la cantidad de coeficientes. Cuanto menor es es el valor AICc mejor será dicho modelo para representar la serie temporal.

A continuación se analizan los coeficientes para verificar que la muestra proviene de variables aleatorias independientes e idénticamente distribuidas, para ello cada coeficiente deberá cumplir la siguiente premisa:

$$\hat{\rho_k} \approx \mathcal{N}(0, \frac{1}{\sqrt{T}})$$

Es decir que aquellos coeficientes cuyo valor no supera 1.96 veces la raíz cuadrada de su varianza harán que se rechace la independencia al 5%.

$$|\hat{\rho_k}| \geq \frac{1.96}{\sqrt{T}}$$

Para poder eliminar los coeficientes se elimina el coeficiente que tenga el menor valor con respecto a su varianza y se calcula de nuevo el ratio de todos los coeficientes, así se van eliminando uno a uno todos aquellos que no cumplen el criterio quedándose finalmente los coeficientes presentes en la figura 4.9 .

ar1	ar2	ar3	ar4	ma1	ma2	intercept
0.0000000	0.0000000	0.6608770	0.1067055	0.7896443	0.6667272	23.6983122

	ar3	ar4	ma1	ma2	intercept
ar3	0.0003113876	$-1.141557e-04$	0.0001529017	0.0002350160	$-3.330842e-04$
ar4	-0.0001141557	$2.196735e-04$	-0.0001378381	-0.0001233695	$-6.742386e-06$
ma1	0.0001529017	$-1.378381e-04$	0.0002553670	0.0001970100	$-1.692062e-04$
ma2	0.0002350160	$-1.233695e-04$	0.0001970100	0.0003310874	$-2.732637e-04$
intercept	-0.0003330842	$-6.742386e-06$	-0.0001692062	-0.0002732637	$8.065207e+00$

Figura 4.9: Coeficientes del modelo ARIMA resultante

Una vez eliminados uno a uno se hace la prueba de nuevo para ver si existe algún coeficiente más que sea susceptible de no cumplir el criterio ver figura 4.10. Llegado a este punto ya se dispone de un modelo que previsiblemente se ajusta a la serie temporal y es susceptible de realizar predicciones una vez validado.

ar3	ar4	ma1	ma2	intercept
FALSE	FALSE	FALSE	FALSE	FALSE

Figura 4.10: Coeficientes cuya distribución cumple la premisa de independencia de un modelo ARIMA. Nota: FALSE indica que la raíz cuadrada de la varianza multiplicada por 1.96 no supera el valor del coeficiente.

Para validar el modelo se deben analizar los residuos verificando que cumplen con las hipótesis básicas sobre las que se sustenta un modelo ARMA y que exigen que los residuos del modelo cumplan con las siguientes características:

- Media cero
- Varianza constante
- Incorreladas

Además de estas características, a la hora de realizar predicciones, también es conveniente que sigan una distribución normal, aunque si no se cumple este requisito no se invalida el modelo.

La primera prueba que se realiza es el test de Ljung-Box que mide la independencia y aleatoriedad de una serie temporal. En caso de que exista dependencia en los residuos, y por lo tanto no sean incorrelados, significa que el modelo no captura todas las relaciones de dependencia y por lo tanto es incompleto. En el test de Ljunb-Box la hipótesis nula establece que los residuos son independientes, como se puede comprobar en la figura 4.11 el $p - valor$ es de 0.001, inferior al 0.05 lo que indica que se rechaza la hipótesis nula. Los residuos son dependientes y por lo tanto el modelo no es válido. Es posible que el modelo no sea lo suficientemente complejo para ajustar la serie temporal (y es por lo tanto un problema del modelo) o que existan factores externos que influyan en la variable de salida y que no dependen de la serie temporal (el caudal de entrada depende de otros factores).

Ljung-Box test

data: Residuals from ARIMA(4,0,2) with non-zero mean

$Q^* = 846.37$, df $= 724$, p-value $= 0.001078$

Model df:6. Total lags used:730

Figura 4.11: Análisis de independencia de los residuos

Análisis de descarga

Debido a que se ha seleccionado un modelo que tiene en cuenta el contexto temporal hay un aspecto que se debe analizar a la hora de manejar los datos. Cuánto tiempo tarda el agua de cada una de las cuencas en llegar al caudal de entrada desde que toca el suelo, esto es lo que se conoce como análisis de descarga y tiene mucha importancia a la hora de poder establecer las ventanas temporales de predicción.

Uno de los aspectos a tener en cuenta en el análisis de descarga es el estado del terreno Meier et al. [41], es por ello que se ha analizado el tiempo transcurrido desde que una gota cae al suelo (se registra en el pluviómetro) hasta que llega al caudal en función de la humedad intrínseca del terreno. Puesto que no se dispone de sensores de humedad en cada una de las cuencas lo que se realiza es un agrupado de los posibles estados del terreno en función de la acumulación de las lluvias pasadas y se calcula el día en el que se obtiene la máxima correlación (R^2) entre eventos de precipitación y variación de caudales de entrada

Primero se crea una nueva variable, esta variable establecerá el estado de humedad del terreno con la información de la que se dispone (cantidad de lluvia pasada). Para ello se utiliza la cantidad de precipitaciones acumuladas en las últimas semanas. Se asume, dado que no se puede comparar con otros trabajos debido a que depende de las características particulares del terreno y la situación de los sensores, que el agua tarda menos de una semana en llegar desde las zonas de precipitación hasta el cauce de entrada de la presa. Por lo tanto lo que determinará el estado del terreno son las semanas anteriores.

Para cada zona, se crea una variable que aglutina la cantidad de agua que ha llovido en esa zona en las semanas -4, -3 y -2. Dejando la semana -1 como agua que llega al cauce del río. La cantidad de precipitación acumulada de la última semana se calcula por separado, en caso de que tardase mucho menos de una semana en llegar el agua al cauce de entrada a la presa la información seguirá siendo integrada al completo.

Una vez se disponen de los sumatorios de agua acumulada en cada una de las cuencas se discretiza el estado del terreno en función de dichas variables, para ello se utiliza un algoritmo KMeans que clasifica cada registro diario en 5 grupos diferentes normalizados por media y desviación típica. El algoritmo KMeans es útil en este caso debido a que los eventos de precipitación no son aislados para cada una de las zonas, por lo tanto la agrupación presumiblemente se realizará por acumulado temporal y no por las diferentes zonas espaciales. Una vez se obtienen los 5 grupos se calcula el sumatorio de las columnas de cada centroide lo que nos devuelve la concentración media normalizada de agua acumulada para cada grupo.

Media Normalizada	**Estado**
1.744361	Saturado
0.805600	Mojado
0.650348	Húmedo
0.282704	Algo Húmedo
-0.454836	Seco

Cuadro 4.1: Tabla de valores medios de precipitación acumulada de los centroides y su etiqueta asignada

Esto permite dividir y discretizar el estado del terreno en función de la humedad. Aquellos grupos con la cantidad acumulada más baja serán lo que representen los eventos de mayor sequía del terreno y aquellos con una mayor cantidad media acumulada serán los que representan los estados de terreno más húmeda. Las etiquetas no representan el estado real del terreno, son solo un indicativo relativo que ayuda a contextualizar cada uno de los centroides.

Cuadro 4.2: Correlación máxima entre el evento de precipitación y la alteración en el caudal de entrada a la presa (en días)

Humedad	Arzua	Olveda	Serradofaro	Melide
Seco	1	1	1	1
Algo húmedo	1	1	1	1
Húmedo	1	1	1	1
Mojado	1	1	1	1
Saturado	1	1	1	1

Como se puede comprobar en zona de precipitación y en cualquier estado del terreno el tiempo transcurrido es siempre inferior a 24h, es decir, el agua tarda menos de 24h en llegar a la cuenca independientemente de la zona en la que precipita (teniendo en cuenta únicamente las cuatro zonas de influencia indicadas anteriormente).

Correlación

Una vez confirmado que no se puede predecir el caudal futuro únicamente con el caudal pasado y teniendo en cuenta que la máxima correlación entre la precipitación y el caudal de entrada se produce en un plazo inferior a un día se procede a hacer un análisis previo de correlación lineal entre las variables de entrada y de salida con la matriz de correlaciones para poder conocer su grado, en caso de que fuese una correlación muy alta cualquier otro análisis perdería importancia debido a que una simple dependencia lineal permitiría definir el sistema. Por lo tanto se calcula el coeficiente de correlación de Pearson de la precipitación en cada una de las zonas de influencia el día anterior y el caudal del día presente.

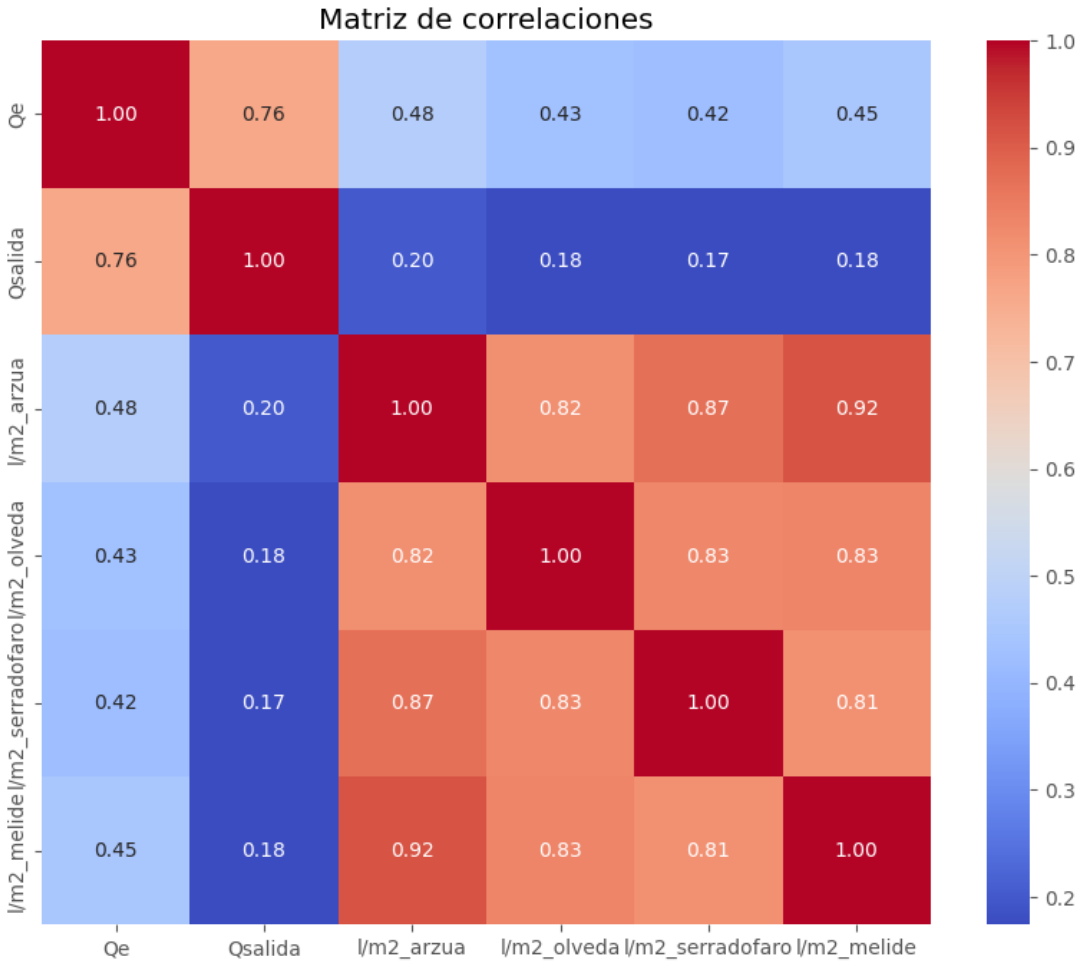

Figura 4.12: Correlaciones entre las diferentes entradas y las salidas

Como se puede comprobar existe cierta relación lineal entre la lluvia y el caudal de entrada y de salida. También se puede ver que esta relación lineal es más fuerte con el caudal de entrada que con el de salida, algo totalmente lógico debido a que existen más variables que influye en la salida de la presa (el criterio del operador) que no hay en la entrada y esta gráfica nos demuestra que dicha relación no es lineal.

Por otra parte se puede constatar que la relación lineal aunque existe no es significativa, lo que muestra que existe influencia de las entradas en la salida, pero dicha influencia es baja debido a que no es en su mayor parte lineal o porque hay otros factores que influyen en la variable de salida.

4.2.3 Preparación de los datos

En una primera instancia se integran en un único dataset todos los registros de las diferentes fuentes, puesto que el rango de fechas disponible en cada uno de los registros es diferente se procede a seleccionar aquellas fechas que están presentes en todos ellos. Una vez se obtiene el dataset que engloba todos los registros se procede al manejo de datos incorrectos o ausentes. Aquellos días cuyo valor no haya sido registrado se eliminan del dataset. Esto ocurre en 149 días de 4554 en total lo que hace un 3.27% del total que es un porcentaje muy bajo para que tenga impacto, sobre todo si se tiene en cuenta que los registros ausentes no están agrupados.

Debido a la naturaleza de los datos (registros de precipitación y caudales) se analiza si algún registro contiene valores negativos (no existen caudales negativos o precipitaciones negativas) que podrían ser debidos a fallos en los sensores así como registros extremadamente altos (caudales o precipitaciones imposibles), no se detecta ningún valor anormalmente alto pero sí varios valores negativos (todos próximos a cero), se asume que dichos valores son un defecto en el registro y se cambian dichas celdas por cero.

Una vez eliminados los registros ausentes y se corrigen los registros anómalos se procede a la integración de las diferentes fuentes (registros de la presa por un lado y registros de precipitaciones por otro).

Selección de variables

Aunque el número de registros disponibles es bastante elevado, el número de características de cada registro es bajo y muchas de ellas se derivan unas de otras, por lo que no se aplica una extracción de variables en este entorno en concreto. El objetivo principal del estudio es poder ofrecer un modelo que pueda predecir el estado futuro de la presa (caudal de entrada y salida) con las variables conocidas en el momento de la toma de decisiones para evitar eventos extremos. Finalmente las variables con las que se trabaja para entrenar los modelos son las siguientes:

- Q_e: Caudal de entrada a la presa en $t-1$
- Q_s: Caudal de salida de la presa (suma de caudal de las tres compuertas de salida en $t-1$
- ℓ/m^2 Arzúa: Precipitación registrada en Arzúa en $t-1$
- ℓ/m^2 Olveda: Precipitación registrada en Olveda en $t-1$
- ℓ/m^2 Serradofaro: Precipitación registrada en Serradofaro en $t-1$

- ℓ/m^2 Melide: Precipitación registrada en Melide en $t-1$
- pred ℓ/m^2: Predicción de Meteogalicia a un día de precipitación global de la zona en $t-1$
- pred ℓ/m^2 2d: Predicción de Meteogalicia a dos días de precipitación global de la zona en $t-2$
- pred ℓ/m^2 3d: Predicción de Meteogalicia a tres días de precipitación global de la zona en $t-3$

Selección de la ventana temporal

Dado que se desconoce cuánto tiempo dura la influencia de las variables de entrada sobre cada una de las salidas, se prueban tres ventanas de contexto diferentes de tres, siete y quince días de contexto. En el estudio de Jo and Jung [15] utilizado como base para este estudio, se utilizan varias ventanas de contexto de 1 a 15 días. Se tiene en cuenta que el tiempo de descarga (que es el tiempo que tarda el agua en llegar a la entrada de la presa desde la precipitación) es inferior a 24 horas, obtenido en el análisis de descarga, por lo que se espera que un periodo de 15 días como máximo sea más que suficiente.

4.2.4 Selección de hiperparámetros

En caso de la red LSTM se ha realizado una búsqueda de pocos hiperparámetros tomando como punto de partida otros trabajos que utilizaron redes LSTM en modelos lluvia-escorrentía Li et al. [48] (número de epochs 500).

Tomando como base una red LSTM de una única capa oculta los valores entre los que se realizó la búsqueda de hiperparámetros es la siguiente (se podrían haber aumentado el número de hiperparámetros tanto en rango de valores como de número de hiperparámetros, pero únicamente por una cuestión de coste temporal se realizaron 8 iteraciones):

- Número de neuronas en la capa oculta
- Normalización por dropout usando un ratio de neuronas eliminadas de un 20% (si/no)
- Normalización después de cada capa (si/no)

Después de realizar la búsqueda de hiperparámetros y seleccionando aquella red que devolvía el menor valor de error cuadrático medio (MSE) en un conjunto de prueba usando el caudal de entrada como variable de salida, se seleccionó una red con las siguientes características. Para realizar dicha selección se dispuso el código de tal manera que para cada experimento

evaluase el MSE en el conjunto de evaluación, se registra su MSE y, si dicho modelo mejora al anterior, se guarda el modelo para su uso posterior.

Neuronas capa oculta	Dropout	Batch normalization	MSE en validación
12	0	0	0.001576
12	**1**	**0**	**0.000806**
12	0	1	0.001444
12	1	1	0.004551
64	0	0	0.001150
64	1	0	0.001250
64	0	1	0.002201
64	1	1	0.013554

Cuadro 4.3: Resultado de cada uno de los modelos de una red LSTM con hiperparámetros diferentes en el conjunto de validación

Como resultado de este análisis, finalmente la red que se utiliza para entrenamiento y predicciones es la siguiente:

- Número de neuronas en la capa oculta: 12
- Normalización por dropout usando un ratio de neuronas eliminadas de un 20% (si/no): Sí
- Normalización después de cada capa (si/no): No

Para el resto de modelos, debido a que se han utilizado para realizar un estudio comparativo con respecto a la red LSTM se utilizaron sus valores por defecto.

Capítulo 5

Pruebas

Estos son los resultados de los diferentes experimentos realizados. Se dividen en dos secciones, uno para las predicciones del caudal de entrada de la presa y otro para la predicción del caudal de salida. A su vez, cada una de las secciones se divide en función de la ventana temporal de contexto. Aunque todas las predicciones son a un día vista se han probado contexto de 3, 7 y 15 días, es decir, que para predecir la salida cada uno de los modelos se ha ejecutado utilizando la información de los 3, 7 y 15 días previos.

5.1 Resultados en la predicción de caudal de entrada

5.1.1 Resultados de predicción de caudal de entrada del día siguiente con el contexto de 3 días previos

Modelo	NS Train	NS Test	MSE Train	MSE Test	R2 Train	R2 Test
LSTM	0.903	0.934	94.124	**75.063**	0.926	0.937
SVR	-0.424	-0.450	895.886	1028.513	0.810	0.643
RF	0.980	**0.937**	21.734	75.277	0.983	**0.938**
XGBoost	**1.000**	0.898	**0.422**	123.493	**1.000**	0.901
ANN	0.821	0.866	184.961	143.551	0.851	0.887
NAIVE	0.773	0.848	272.903	180.106	0.786	0.854

Cuadro 5.1: Evaluación de la predicción en el caudal de entrada del conjunto de test con 3 días de contexto, predicción del día siguiente.

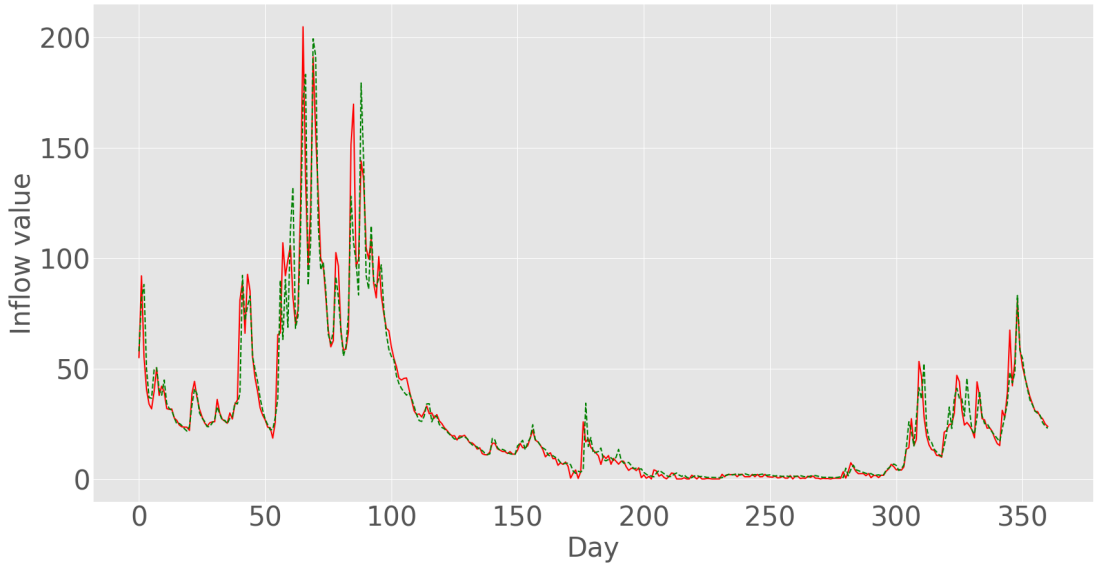

Figura 5.1: Predicciones de la red LSTM (verde) frente a los valores reales (rojo) del conjunto de entrenamiento. Contexto 3 días.

En general todos los algoritmos obtienen unos valores de NS muy altos a excepción del algoritmo de SVR. RF obtiene el mejor resultado logrando tres centésimas por encima de la red LSTM, estas dos son las únicas que obtienen valores por encima del 0.9. Se puede ver claramente sobreentrenamiento en el caso de XGBoost y posiblemente también en RF.

5.1.2 Resultados de predicción de caudal de entrada del día siguiente con el contexto de 7 días previos

Modelo	NS Train	NS Test	MSE Train	MSE Test	R2 Train	R2 Test
LSTM	0.903	**0.944**	92.716	**61.883**	0.929	**0.948**
SVR	0.590	0.331	404.418	716.518	0.804	0.612
RF	0.982	0.930	19.824	84.410	0.984	0.931
XGBoost	**1.000**	0.911	**0.212**	107.875	**1.000**	0.912
ANN	0.783	0.849	183.047	131.039	0.856	0.899
NAIVE	0.772	0.853	272.551	173.853	0.785	0.858

Cuadro 5.2: Evaluación de la predicción en el caudal de entrada del conjunto de test con 7 días de contexto, predicción del día siguiente

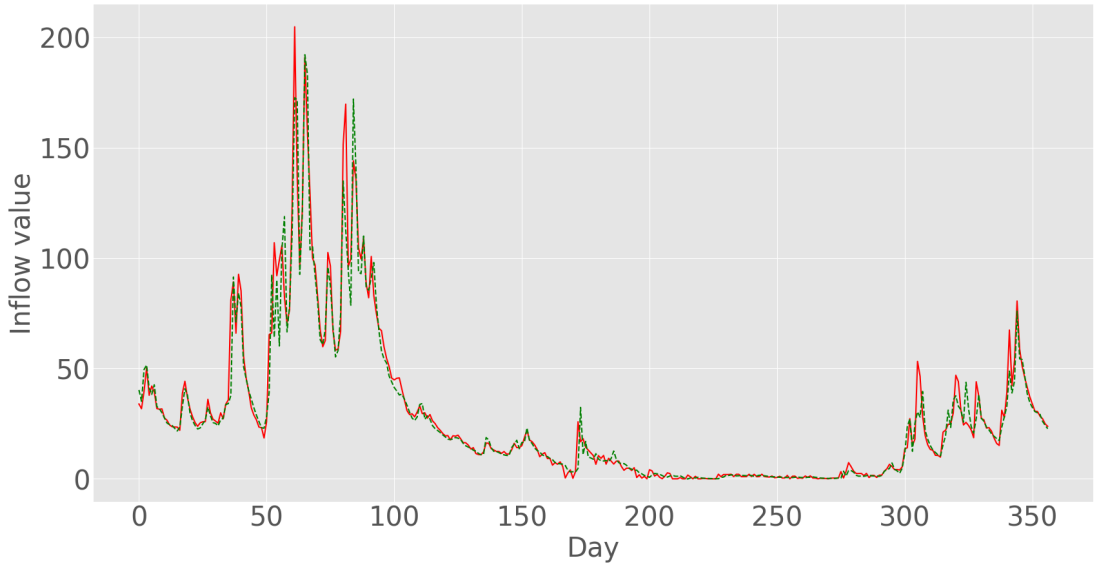

Figura 5.2: Predicciones de la red LSTM (verde) frente a los valores reales (rojo) del conjunto de entrenamiento.Contexto 7 días.

Al aumentar la ventana temporal la información extra mejora aun más los resultados del algoritmo LSTM mientras que el resto se mantienen más o menos igual, se aprecia que aunque la información en la entrada es mayor el sobreentrenamiento en el caso de XGBoost sigue siendo muy claro y en RF se puede intuír el mismo comportamiento que en el caso anterior.

5.1.3 Resultados de predicción de caudal de entrada del día siguiente con el contexto de 15 días previos

Modelo	NS Train	NS Test	MSE Train	MSE Test	R2 Train	R2 Test
LSTM	0.899	**0.950**	96.412	**59.213**	0.922	**0.951**
SVR	0.693	0.476	420.410	757.401	0.749	0.609
RF	0.981	0.933	20.137	79.510	0.984	0.935
XGBoost	**1.000**	0.919	**0.143**	111.579	**1.000**	0.921
ANN	0.794	0.857	199.334	153.159	0.853	0.898
NAIVE	0.774	0.853	266.826	176.521	0.787	0.859

Cuadro 5.3: Evaluación de la predicción en el caudal de entrada del conjunto de test con 15 días de contexto, predicción del día siguiente

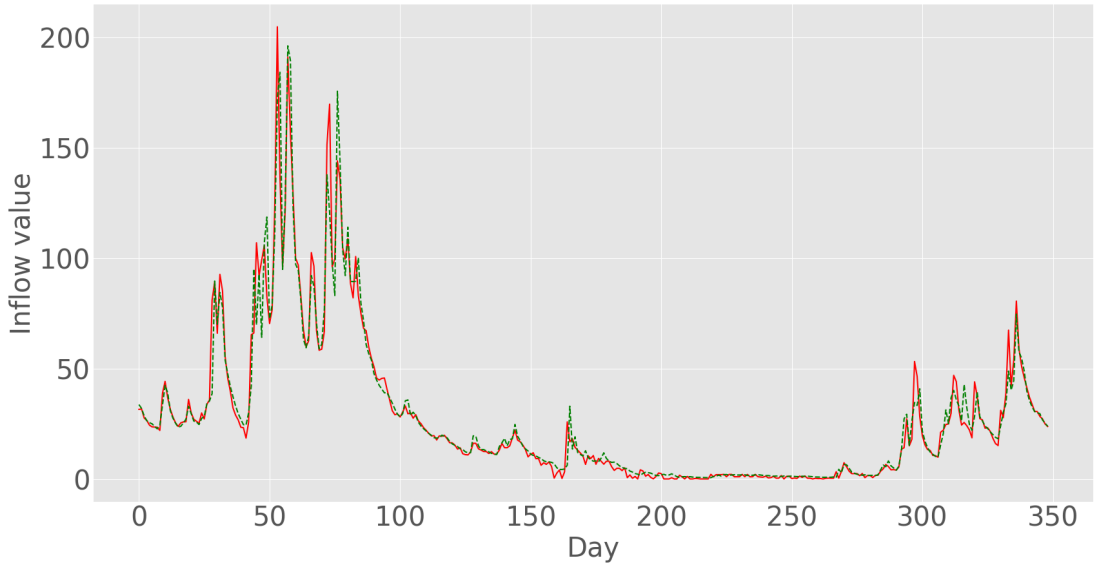

Figura 5.3: Predicciones de la red LSTM (verde) frente a los valores reales (rojo) del conjunto de entrenamiento. Contexto 15 días.

En este caso se sigue la tendencia anterior, aumentar la ventana de contexto permite al algoritmo LSTM mejorar en sus predicciones (generalizar mejor) a pesar de obtener un valor similar en el conjunto de entrenamiento. Este aumento de información en la entrada no mitiga las causas que originan el sobreentrenamiento.

5.2 Resultados en la predicción de caudal de salida

5.2.1 Resultados de predicción de caudal de salida del día siguiente con el contexto de 3 días previos

Modelo	NS Train	NS Test	MSE Train	MSE Test	R2 Train	R2 Test
LSTM	0.909	**0.900**	60.287	**64.580**	0.918	**0.910**
SVR	0.147	0.055	294.170	349.958	0.816	0.750
RF	0.988	0.886	8.431	77.116	0.989	0.893
XGBoost	**0.999**	0.891	**0.769**	77.654	**0.999**	0.895
ANN	0.885	0.855	67.619	82.255	0.912	0.891
NAIVE	0.899	0.887	74.397	80.598	0.901	0.890

Cuadro 5.4: Evaluación de la predicción en el caudal de salida del conjunto de test con 3 días de contexto, predicción del día siguiente

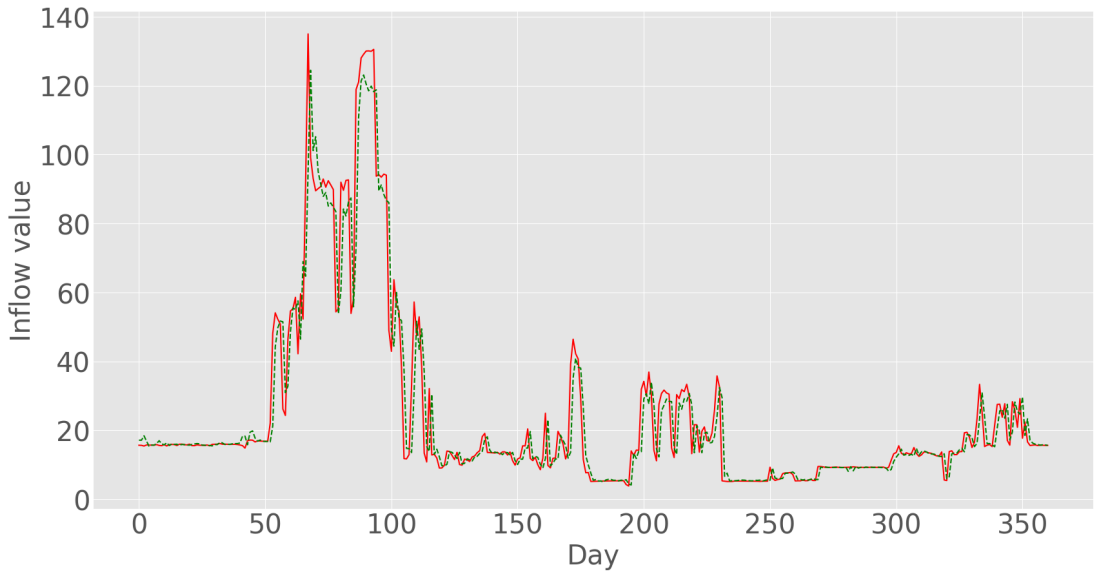

Figura 5.4: Predicciones de la red LSTM (verde) frente a los valores reales (rojo) del conjunto de entrenamiento. Contexto 3 días.

En el caso de la predicción de salida para un contexto de 3 días se obtiene un resultado muy similar al Naive por parte de prácticamente todos los algoritmos a excepción del SVR. El sobreentrenamiento en caso de XGBoost y RF parece que sigue presente y de nuevo la red LSTM es la que mejor resultado da.

Modelo	NS Train	NS Test	MSE Train	MSE Test	R2 Train	R2 Test
LSTM	0.905	**0.894**	58.358	**66.293**	0.921	**0.909**
SVR	0.420	0.162	217.859	328.855	0.836	0.683
RF	0.988	0.889	8.240	76.036	0.989	0.896
XGBoost	**1.000**	0.875	**0.356**	87.622	**1.000**	0.881
ANN	0.893	0.865	65.320	81.083	0.913	0.888
NAIVE	0.898	0.887	74.231	81.501	0.900	0.890

Cuadro 5.5: Evaluación de la predicción en el caudal de salida del conjunto de test con 7 días de contexto, predicción del día siguiente

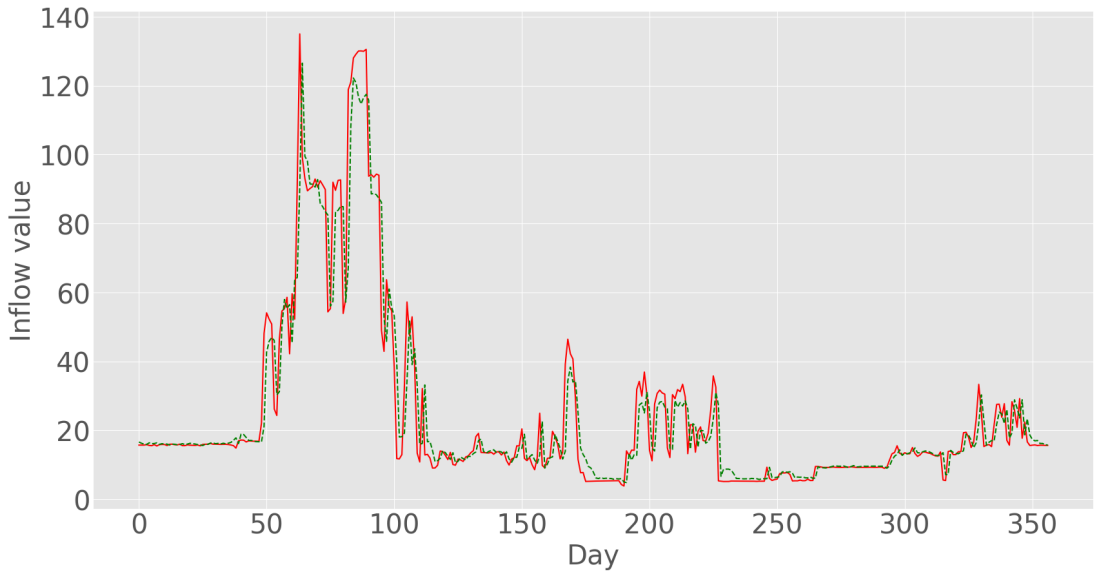

Figura 5.5: Predicciones de la red LSTM (verde) frente a los valores reales (rojo) del conjunto de entrenamiento. Contexto 7 días.

En este caso el aumentar el contexto no mejora los resultados, todos los algoritmos dan resultados muy similares permaneciendo las métricas de manera estable.

5.2.2 Resultados de predicción de caudal de salida del día siguiente con el contexto de 15 días previos

Modelo	NS Train	NS Test	MSE Train	MSE Test	R2 Train	R2 Test
LSTM	0.906	**0.894**	60.336	**70.400**	0.917	**0.905**
SVR	0.516	0.246	200.321	332.532	0.819	0.666
RF	0.988	0.883	8.255	82.491	0.989	0.889
XGBoost	**1.000**	0.862	**0.208**	97.144	**1.000**	0.870
ANN	0.893	0.856	67.374	93.174	0.915	0.887
NAIVE	0.897	0.886	74.082	83.368	0.900	0.890

Cuadro 5.6: Evaluación de la predicción en el caudal de salida del conjunto de test con 15 días de contexto, predicción del día siguiente

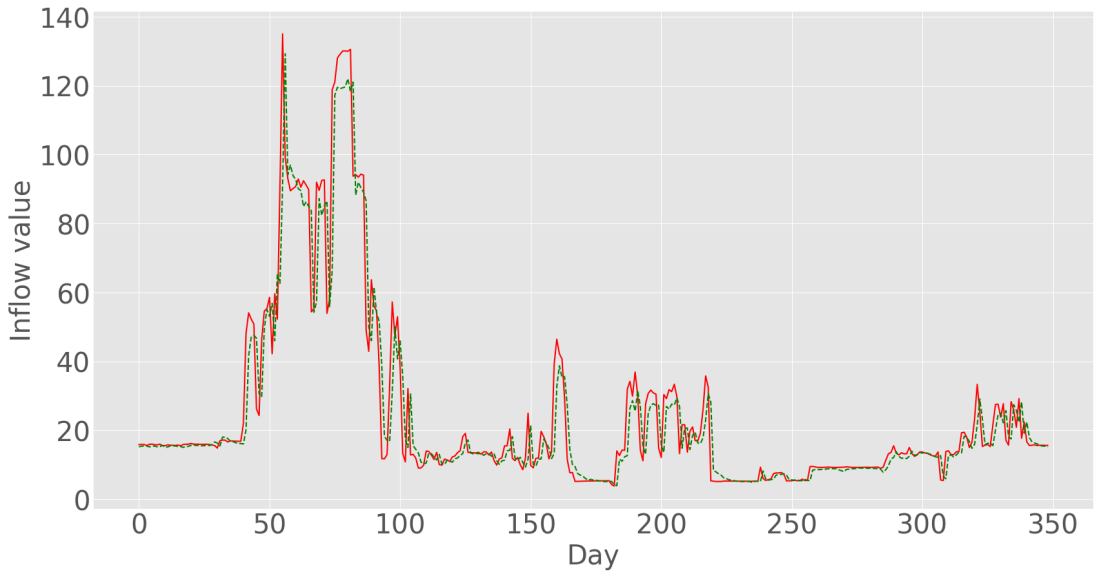

Figura 5.6: Predicciones de la red LSTM (verde) frente a los valores reales (rojo) del conjunto de entrenamiento. Contexto 15 días.

Del mismo modo que ocurrió en el caso anterior los valores permanecen prácticamente idénticos, el aumentar el tamaño de la ventana no mejora los resultados y tampoco mitiga los efectos del sobreentrenamiento en XGBoost y RF.

Capítulo 6

Conclusiones

Como se puede observar en el cuadro 5.3, en la predicción de un día y una ventana de contexto de 15 días la red LSTM alcanza una puntuación en test de 0.95 para el Coeficiente de Eficiencia del Modelo de Nash Sutcliffe (NS). Estos valores de NS son mejores que los encontrados en otros estudios, como 0.857 con un contexto de 10 días y 0.839 con un contexto de 30 días en el conjunto de test por Jo and Jung [15] o un rango entre 0.8 y 0.9 en test por Dongkyun and Seokkoob [14] y muy similar al obtenido en test por Li et al. [48] que obtuvo un valor de NS de 0.942 pero usando 153 pluviómetros en lugar de 4 como es en este caso.

Estos resultados confirman la robustez de este algoritmo en este conjunto de datos y lo establece como una base válida para futuros ajustes finos o como complemento a otro tipo de algoritmos.

Cabe destacar que la granularidad de los registros son a un día, esto puede penalizar aquellos algoritmos que usan una componente temporal como las redes LSTM si en la variable de salida tienen influencia variables en una escala temporal más fina. Es posible que con registros con un intervalo temporal entre registros más pequeño la red LSTM pueda mejorar los resultados obtenidos. Del mismo modo hay que tener en cuenta que los registros están muy desbalanceados, los eventos de avenida son muy escasos con respecto al resto de los días, es por eso que un entrenamiento ponderado para darle más peso a dichos eventos de interés podría ser también una mejora para afinar la calidad de las predicciones de este modelo.

Con respecto al caudal de salida hay que tener en cuenta un elemento muy importante, dentro del comportamiento de esta variable hay un factor anómalo para modelar, el comportamiento humano. Es el operario de la planta el que en última instancia, en función del estado de las variables de las que tiene constancia en cada momento, elige abrir las compuertas hasta un determinado punto y a pesar de existir unas reglas de operación es la toma de decisiones

de una persona la que modela los patrones de la variable de salida.

Añadido a este punto como se ha visto anteriormente, el operario se rige en parte por unas reglas de operación las cuales le permiten un margen de maniobrabilidad (como se pudo observar en las diferencias entre el caudal teórico y el caudal real). Esto último es lo que podría explicar que el algoritmo permanezca invariable al contexto anterior. Es probable que el comportamiento del operario venga dado por unas reglas propias derivadas de las reglas de operación junto con su propia experiencia que hayan dado como resultado una relación más o menos lineal entre ciertas variables y la variable de salida. Esto también explicaría por que el resultado de los modelos no mejora significativamente el modelo naive. Otra posibilidad es que en este caso el operario se rija del mismo modo que dicha regla, uso del valor de caudal de salida del día anterior para calcular el valor del día siguiente con pequeñas variaciones.

Como conclusión final se puede observar que los algoritmos de aprendizaje automático son una muy buena herramienta para predecir el caudal de entrada. Por lo que puede ser un punto de partida muy interesante para usar estos resultados como núcleo de un sistema de recomendación que ayude al operario a tomar decisiones sabiendo qué es lo que va a pasar en el futuro próximo con un porcentaje de acierto por encima del 90%

Este trabajo fin de grado ha permitido crear una línea de proyecto que se ha presentado en diversos congresos tanto nacionales como internacionales Fernandez et al. [49] Fernandez et al. [50] Fernandez et al. [51] Fernandez et al. [52] (más información en el Apéndice A).

Capítulo 7

Desarrollo Futuro

El establecer una herramienta que permita la predicción de los estados futuros tanto de entrada como de salida en una presa hidroeléctrica permite establecer una base para la gestión más eficiente y sobre todo más segura de este tipo de instalaciones. Pero existe un gran margen de mejora en dicha gestión si a dicha base se le añaden una serie de características que permitirán dar una cobertura más amplia y más precisa a dicha ayuda a la toma de de decisiones en la operativa de una central hidroeléctrica.

En primer lugar, un aumento en el margen de predicción amplia notablemente la posibilidad de minimizar daños o maximizar la eficiencia de un bien escaso como es el agua. En la actualidad se conocen diversas técnicas y estudios que permiten realizar proyecciones del clima a más largo plazo (incluso meses como los trabajos basados en ríos atmosféricos)

En segundo lugar, el poder realizar un ajuste más fino tanto de los hiperparámetros como del propio proceso de entrenamiento. Hay que tener en cuenta que, tal y como se indica en el apartado de conclusiones, existe un desbalanceo muy grande entre días normales y días de eventos extremos (desbordamientos), esto hace que el entrenamiento de un modelo de aprendizaje automático sea mucho más complicado. Entre las soluciones que se pueden proponer está la de aumentar la cantidad de registros (difícil de conseguir) tanto en rango de fechas como en la granularidad de los mismos. Otro mecanismo para poder lidiar con este tipo de situaciones sería el de ponderar los casos. Se puede agregar al algoritmo de entrenamiento un coeficiente de regulación de los gradientes de los pesos en función de la varianza de los últimos ”n” días. Los eventos de lluvia extrema tienen la característica de que son breves y suelen estar precedidos de períodos inestables, cuya varianza es elevada. En cambio los eventos normales suelen ser mucho más estables. Si se multiplica el valor del gradiente de una iteración por la varianza asociada a los períodos anteriores de los eventos que ha procesado permitirá mejorar mucho la dirección hacia la que se pretende llevar al algoritmo. Se penalizaría de este modo

aquellos eventos no relevantes dando mucha mayor importancia (y de manera ponderada) a las regiones más variables.

Otro aspecto crucial es la capacidad del modelo para generar predicciones robustas, fiables y explicables. Un elemento fundamental de los sistemas de soporte a decisiones no es solo sugerir acciones basadas en el estado del sistema y la información disponible, sino también, en muchos casos, proporcionar no solo el 'qué', sino también el 'por qué'. Un ejemplo evidente sería el de los sistemas integrados en el sector sanitario.

Para ello el tipo de algoritmos a elegir no deben basarse únicamente en los resultados obtenidos sino también en la capacidad que tienen dichos algoritmos en proporcionar información acerca de cuales son las variables que más influencia tienen o a qué nivel están influyendo. El dotar de dicha información no solo daría más solidez a cada una de las predicciones sino que permitiría conocer más a fondo como funciona de manera interna este tipo de instalaciones lo que permitiría mejorar su gestión de base.

Este proyecto genera una serie de modelos de predicción para los estados futuros del sistema, pero no genera ningún tipo de información acerca de la toma de acciones necesarias para reducir los riesgos o mejorar la eficiencia. Es por ello que un aspecto clave a la hora de continuar con este proyecto es el de utilizar las predicciones generadas por el sistema para introducirlas en otro algoritmo que minimice los eventos extremos (sobre todo el de avenidas) y maximice la gestión de un bien escaso como es el agua.

Apéndices

Apéndice A

Presentaciones

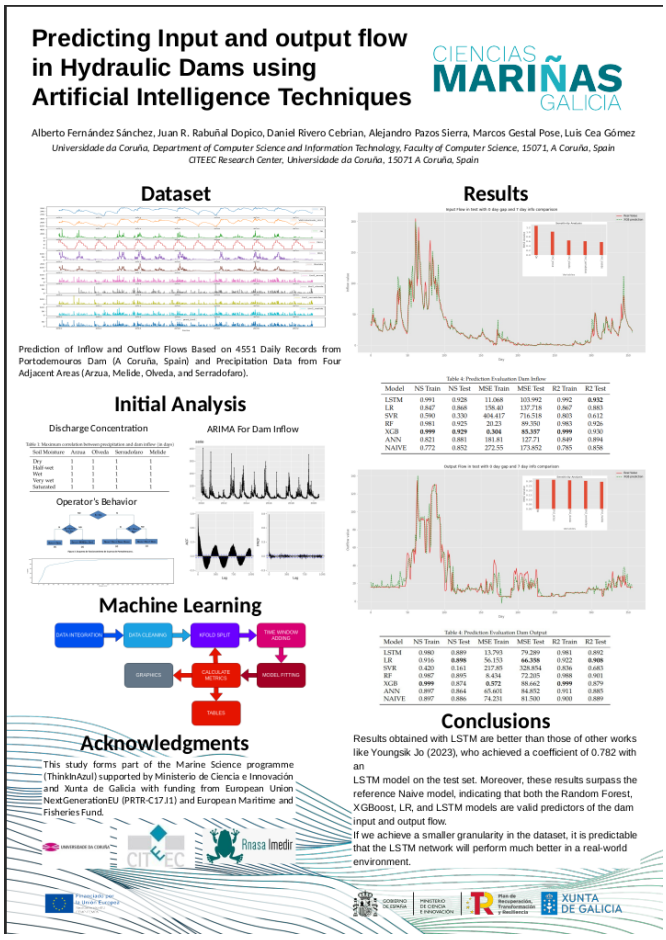

Figura A.1: Poster de la la 3^a asamblea general del programa de ciencias mariñas

Figura A.2: Congreso Bangkok 2023

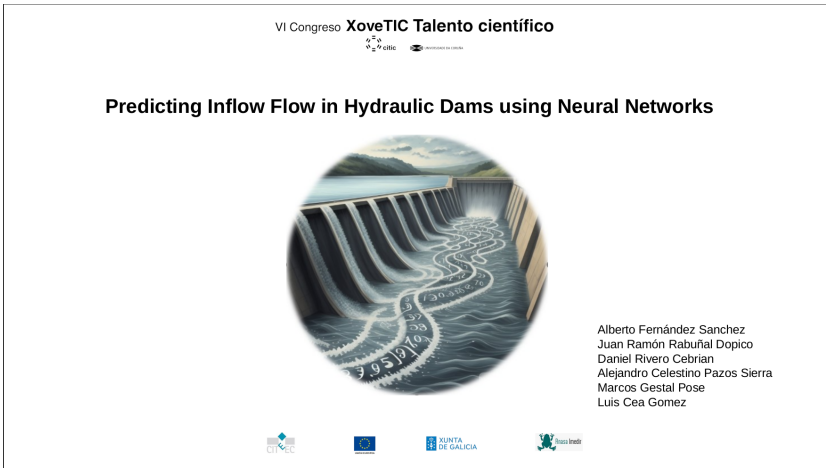

Figura A.3: Xovetic 2023

Figura A.4: Premio concedido a este proyecto en el congreso Xovetic 2023

Apéndice B

Código del proyecto

B.1 Código del proyecto

Todo el código del proyecto está en el repositorio siguiente
https://github.com/AlbertoGCID/CITEEC209
Junto a ese código hay más archivos fruto del proyecto de investigación del que parte este trabajo y que se presentó en diferentes eventos

Apéndice C

Flow predictor tool

C.1 Herramienta de predicción de caudal

Como añadido al trabajo analítico de este estudio se ha desarrollado una herramienta para que cualquier usuario pueda realizar sus propias predicciones utilizando sus propios datasets. Esta herramienta automatiza todo el proceso de integración, preprocesado y predicción con la posibilidad de reentrenar con un dataset diferente siempre y cuando contenga las mismas características que el original.

https://github.com/AlbertoGCID/flowpredictor

```
└── src
    ├── AI_algorithms.py
    ├── AI_results.py
    ├── __init__.py
    ├── data_load.py
    ├── main.py
    ├── split_delay_time.py
    ├── test.py
    ├── models
    │   ├── inputLSTM.keras
    │   └── outputLSTM.keras
    └── scaler
        ├── inflow_train_scaler.pkl
        ├── outflow_train_scaler.pkl
        └── x_train_scaler.pkl
```

C.2 Overview

This project provides a $RainfallDataset$ class and utility functions for time series prediction using Long Short-Term Memory (LSTM) networks. The main function $main()$ serves as a script for training and prediction. The script is designed to be run from the command line, accepting various arguments for configuration.

C.3 Prerequisites

Make sure you have the following installed:

- Python 3.10.9
- Required libraries: pandas, numpy, scikit-learn, tensorflow (or keras), prettytable, hydroeval

Install the dependencies using the following command:

```
pip install -r requirements.txt
```

C.4 Usage

C.4.1 Main Function - `main()`

The `main()` function is the entry point for the training and prediction script. It accepts command-line arguments to configure the training and prediction process.

```
python main.py --train_folder datasets/train_folder/
    --predict_folder datasets/predict_folder/ --input_width 7
    --offset 3 --inflow_name input --outflow_name output --target
    input --model_path models/inflowLSTM.keras --x_scaler_path
    scaler/x_train_scaler.pkl --inflow_scaler_path
    scaler/inflow_train_scaler.pkl --outflow_scaler_path
    scaler/outflow_train_scaler.pkl
```

C.4.2 Command-Line Arguments

- **–train_folder**: Path to the training dataset folder
- **–predict_folder**: Path to the prediction dataset folder
- **–input_width**: Width of the input window in days.

- **–offset**: Offset between input and output in days.
- **–inflow_name**: Name of the input column for inflow.
- **–outflow_name**: Name of the output column for outflow.
- **–target**: Name of the target column.
- **–predict_only**: Run prediction only (flag).
- **–model_path**: Path to the trained model (optional for prediction).
- **–x_scaler_path**: Path to the **x_scaler** (optional for prediction).
- **–inflow_scaler_path**: Path to the **inflow_scaler** (optional for prediction).
- **–outflow_scaler_path**: Path to the **outflow_scaler** (optional for prediction).

C.5 Example Usage

C.5.1 Training

```
python main.py --train_folder datasets/train_folder/ --input_width
    7 --offset 3 --inflow_name input --outflow_name output --target
    input
```

C.5.2 Prediction only

```
python main.py --predict_folder datasets/predict_folder/
    --input_width 7 --offset 3 --inflow_name input --outflow_name
    output --target input --predict_only --model_path
    models/inflowLSTM.keras --x_scaler_path
    scaler/x_train_scaler.pkl --inflow_scaler_path
    scaler/inflow_train_scaler.pkl --outflow_scaler_path
    scaler/outflow_train_scaler.pkl
```

Bibliografía

[1] P. Schober, C. Boer, and L. A. Schwarte, "Correlation coefficients: Appropriate use and interpretation." *Anesthesia Analgesia*, vol. 126, no. 5, pp. 1763–1768, 2018.

[2] S. M. Vallejo-Bernal, S. M. Vallejo-Bernal, J. M. Ramirez, and G. Poveda, "A conceptual stochastic rainfall-runoff model of an order-one catchment under a stationary precipitation regime," *Stochastic Environmental Research and Risk Assessment*, vol. 35, no. 11, pp. 1–26, 2021.

[3] F. F. Snyder, "Synthetic unit graphs," *Eos, Transactions American Geophysical Union*, vol. 19, no. 1, pp. 447–454, 1938.

[4] V. T. Chow, D. R. Maidment, and L. W. Mays, *Applied hydrology*. McGraw-Hill Company, 1988.

[5] I. Rodriguez-Iturbe and A. Rinaldo, *Fractal River Basins: Chance and Self-Organization*. Cambridge University Press, 1997.

[6] J. S. Smart, "Channel networks," *Advances in Hydroscience*, vol. 8, pp. 305–346, 1972.

[7] A. B. Taylor and H. E. Schwarz, "Unit-hydrograph lag and peak flow related to basin characteristics," *Transactions, American Geophysical Union*, vol. 33, no. 2, pp. 235–246, 1952.

[8] Soil Conservation Service, "Urban hydrology for small watersheds," US Department of Agriculture, Washington, D.C., Technical Release 55, 1986.

[9] J. E. Nash, "The form of the instantaneous unit hydrograph," *Publications—International Association of Hydrological Sciences*, vol. 45, pp. 114–121, 1957.

[10] H. E. Beck, A. I. J. M. van Dijk, A. de Roo, E. Dutra, G. Fink, R. Orth, and J. Schellekens, "Global evaluation of runoff from 10 state-of-the-art hydrological models," *Hydrology and Earth System Sciences*, vol. 21, no. 6, pp. 2881–2903, 2017, Último acceso: 10 de junio de 2024. [En línea]. Disponible en: https://hess.copernicus.org/articles/21/2881/2017/

[11] L. Ciabatta, L. Brocca, C. Massari, T. Moramarco, S. Gabellani, S. Puca, and W. Wagner, "Rainfall-runoff modelling by using sm2rain-derived and state-of-the-art satellite rainfall products over italy," *International Journal of Applied Earth Observation and Geoinformation*, vol. 48, pp. 163–173, 2016, Último acceso: 10 de junio de 2024. [En línea]. Disponible en: https://www.sciencedirect.com/science/article/pii/S0303243415300374

[12] M. Heřmanovský, V. Havlíček, M. Hanel, and P. Pech, "Regionalization of runoff models derived by genetic programming," *Journal of Hydrology*, vol. 547, pp. 544–556, 2017, Último acceso: 10 de junio de 2024. [En línea]. Disponible en: https://www.sciencedirect.com/science/article/pii/S0022169417300951

[13] J. R. Rabuñal, J. Puertas, J. Suárez, and D. Rivero, "Determination of the unit hydrograph of a typical urban basin using genetic programming and artificial neural networks," *Hydrological Processes*, vol. 21, no. 4, pp. 476–485, 2007, Último acceso: 10 de junio de 2024. [En línea]. Disponible en: https://onlinelibrary.wiley.com/doi/abs/10.1002/hyp.6250

[14] K. Dongkyun and K. Seokkoob, "Data collection strategy for building rainfall-runoff lstm model predicting daily runoff," *J. Korea Water Resour. Assoc.*, vol. 54, no. 10, pp. 795–805, 2021.

[15] Y. Jo and K. Jung, "Comparative study of machine learning and deep learning models applied to data preprocessing methods for dam inflow prediction," *GeoAI Data Society*, vol. 5, no. 2, pp. 92–102, 2023.

[16] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, Último acceso: 10 de junio de 2024. [En línea]. Disponible en: https://doi.org/10.1038/s41586-020-2649-2

[17] T. pandas development team, "pandas-dev/pandas: Pandas," Feb. 2020, Último acceso: 10 de junio de 2024. [En línea]. Disponible en: https://doi.org/10.5281/zenodo.3509134

[18] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[19] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore,

D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Último acceso: 10 de junio de 2024. [En línea]. Disponible en: https://www.tensorflow.org/

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[21] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2021, Último acceso: 10 de junio de 2024. [En línea]. Disponible en: https://www.R-project.org/

[22] GitHub, "Github," 2020, Último acceso: 10 de junio de 2024. [En línea]. Disponible en: https://github.com/

[23] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, fifth edition ed., ser. Wiley Series in Probability and Statistics. Wiley, 2016.

[24] S. Schnabel, "The role of interannual rainfall variability on runoff generation in a small dry sub-humid watershed with disperse tree cover," vol. 39, no. 2, pp. 259–285, 2013.

[25] S. Thiesen, P. Darscheid, and U. Ehret, "Identifying rainfall-runoff events in discharge time series: a data-driven method based on information theory," *Hydrology and Earth System Sciences*, vol. 23, no. 2, pp. 1015–1034, 2019.

[26] V. Havlíček, M. Hanel, P. Máca, M. Kuráž, and P. Pech, "Incorporating basic hydrological concepts into genetic programming for rainfall-runoff forecasting," *Springer-Verlag*, vol. 95, no. 1, pp. 363–380, 2013.

[27] M. B. Ansori and N. Anwar, "The trmm rainfall-runoff transformation model using gr4j as a prediction of the tugu dam reservoir inflow," *GEOMATE Journal*, vol. 23, no. 97, p. 45–52, 2022, Último acceso: 10 de junio de 2024. [En línea]. Disponible en: https://geomatejournal.com/geomate/article/view/1975

[28] W. Zhong, R. Li, Y. Q. Liu, and J. Xu, "Effect of different areal precipitation estimation methods on the accuracy of a reservoir runoff inflow forecast model," *IOP Conference Series: Earth and Environmental Science*, vol. 208, no. 1, p. 012043,

dec 2018, Último acceso: 10 de junio de 2024. [En línea]. Disponible en: https://dx.doi.org/10.1088/1755-1315/208/1/012043

[29] M. Amirreza, D. Amirhossein, S. Gerrit, and T. Massoud, "Daily reservoir inflow forecasting using weather forecast downscaling and rainfall-runoff modeling," *Journal of Hydrology: Regional Studies*, vol. 44, no. 101228, pp. 1–20, 2022.

[30] L. Vargas-Garay, O. D. Torres-Goyeneche, and G. A. Carrillo-Soto, "Evaluation of scs - unit hydrograph model to estimate peak flows in watersheds of norte de santander," *Respuestas journal of engineering sciences*, vol. 24, no. 1, pp. 6–15, 2018.

[31] F. Fan, Y. Deng, X. Hu, and Q. Weng, "Estimating composite curve number using an improved scs-cn method with remotely sensed variables in guangzhou, china," *Remote Sensing*, vol. 5, no. 3, pp. 1425–1438, 2013, Último acceso: 10 de junio de 2024. [En línea]. Disponible en: https://www.mdpi.com/2072-4292/5/3/1425

[32] P. Costabile, C. Costanzo, D. Ferraro, F. Macchione, and G. Petaccia, "Performances of the new hec-ras version 5 for 2-d hydrodynamic-based rainfall-runoff simulations at basin scale: Comparison with a state-of-the art model," *Water*, vol. 12, no. 9, 2020, Último acceso: 10 de junio de 2024. [En línea]. Disponible en: https://www.mdpi.com/2073-4441/12/9/2326

[33] A. R. A.R, S. L.M, B. H., A. J.L., J. K., and S. S.K, "Reservoir sediment inflow prediction using integrated rainfall-runoff and discharge–sediment model," *International Journal of Engineering Technology*, vol. 7, no. 4, pp. 917,923, 2018.

[34] X. Zhang, Y. Lu, G. Zhu, X. Wu, D. Zhao, and B. Duan, "Annual runoff forecast based on a combined EEMD-ARIMA model," *Water Supply*, vol. 22, no. 8, pp. 6807–6820, 07 2022, Último acceso: 10 de junio de 2024. [En línea]. Disponible en: https://doi.org/10.2166/ws.2022.262

[35] M. Valipour, "Long-term runoff study using sarima and arima models in the united states," *Meteorological Applications*, vol. 22, no. 3, pp. 592–598, 2015.

[36] A. Pooja Verma and B. Swastika Chakraborty, "Performance estimation of arima model for orographic rainfall region," in *2020 URSI Regional Conference on Radio Science (URSI-RCRS)*, 2020, pp. 1–4.

[37] J. Zhao *et al.*, "Rainfall study based on arima-rbf combined model," *Journal of Physics: Conference Series*, vol. 2294, p. 012029, 2022.

[38] V. Dhote, S. Mishra, J. P. Shukla, and S. K. Pandey, "Runoff prediction using big data analytics based on arima model," *Indian Journal of Geo Marine Sciences*, vol. 47, no. 11, pp. 2163–2170, November 2018, received 20 February 2017; revised 02 June 2017.

[39] P. Waldmann, "On the use of the pearson correlation coefficient for model evaluation in genome-wide prediction," *Frontiers in Genetics*, vol. 10, p. 899, 2019.

[40] H. Razmkhah, A. M. AkhoundAli, F. Radmanesh, and B. Saghafian, "Evaluation of rainfall spatial correlation effect on rainfall-runoff modeling uncertainty, considering 2-copula," *Arabian Journal of Geosciences*, vol. 9, no. 323, pp. 1–12, 2016, published online: 12 April 2016.

[41] P. Meier, A. Fromelt, and W. Kinzelbach, "Hydrological real-time modelling in the zambezi river basin using satellite-based soil moisture and rainfall data," *Hydrology and Earth System Sciences*, vol. 15, no. 3, pp. 999–1008, 2011, Último acceso: 10 de junio de 2024. [En línea]. Disponible en: https://hess.copernicus.org/articles/15/999/2011/

[42] A. Aytek, M. Asce1, and M. Alp, "An application of artificial intelligence for rainfall–runoff modeling," *J. Earth Syst. Sci*, vol. 117, no. 2, pp. 145–155, 2008.

[43] S. S. Kolekar, S. Gite, B. Pradhan, and K. Kotecha, "Behavior prediction of traffic actors for intelligent vehicle using artificial intelligence techniques: A review," *IEEE Access*, vol. 9, pp. 135 034–135 058, 2021.

[44] P. A. Gloor, *AI-based interaction analysis between humans (and other living creatures)*. Edward Elgar Publishing eBooks, 2022.

[45] C. Basavaraj, A. Pyarelal, and E. Carter, "Multi-timescale modeling of human behavior," *arXiv.org*, vol. abs/2211.09001, 2022.

[46] M. Robila and S. A. Robila, "Applications of artificial intelligence methodologies to behavioral and social sciences," *Journal of Child and Family Studies*, vol. 29, no. 10, pp. 2954–2966, 2020.

[47] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, *CRISP-DM 1.0: Step-by-step data mining guide*, 2000.

[48] W. Li, A. Kiaghadi, and C. N. Dawson, "High Temporal Resolution Rainfall Runoff Modelling Using Long-Short-Term-Memory (LSTM) Networks," *arXiv e-prints*, p. arXiv:2002.02568, 2020, Último acceso: 10 de junio de 2024.

[49] A. Fernandez, J. R. Rabuñal, L. Cea, and M. Gestal, "Predicción de caudal de entrada en presas hidráulicas usando redes neuronales," Universidade da Coruña, CITIC, Universidade da Coruña, october 5-6 2023, presentado en el VI Congreso XoveTIC celebrado en el CITIC de la Universidade da Coruña los días 5 y 6 de octubre de 2023.

[50] A. Fernandez, J. R. Rabuñal, M. Gestal, and L. Cea, "Predicción de caudal de entrada y salida en presas hidráulicas usando redes neuronales," Bangkok, Thailand, november 13-14 2023, presentado en la 2nd World Conference on Engineering, Technology and Applied Science durante los días 13 y 14 de noviembre de 2023 en Bangkok, Tailandia.

[51] A. Fernandez, J. Rabuñal, L. Cea, and M. Gestal, "Predicción de caudal de entrada en presas hidráulicas usando algoritmos de inteligencia artificial," Santiago de Compostela, España, october 28 2023, presentado en la 3ª Asamblea General del Programa de Ciencias Mariñas celebrada en Santiago de Compostela el 28 de octubre de 2023.

[52] A. Fernandez, J. R. R. Dopico, L. C. Gómez, and M. G. Pose, "Predicción de la transformación lluvia-escorrentía para la entrada de una presa hidráulica usando redes de neuronas artificiales lstm," MAEB, A Coruña, España, june 19-21 2024, presentado en la XX Conferencia de la Asociación Española para la Inteligencia Artificial, celebrada en A Coruña del 19 al 21 de junio de 2024.