



Facultad de Ciencias

**Desagregación temporal de la
precipitación mediante Redes
Neuronales Generativas Adversarias
(GANs)**

(Temporal disaggregation of precipitation
using Generative Adversarial Neural
Networks (GANs))

Trabajo de Fin de Máster
para acceder al

MÁSTER EN Data Science

Autor: Pablo Herreros Fuentevilla

Director\es: Javier Díez Sierra

Septiembre - 2023

Resumen

En este proyecto se ha estudiado la generación de mapas horarios de precipitaciones a partir de mapas diarios utilizando Redes Neuronales Generativas Adversarias (GANs). El objetivo es analizar si utilizando GANs se puede simular mapas de precipitaciones horarios conservando la estructura de las precipitaciones a dicha resolución. Esto es conocido como desagregación temporal y es de gran utilidad para los casos donde se dispone únicamente de datos diarios y se quiere estudiar a una resolución temporal mayor (horaria) o para mejorar la caracterización de la lluvia con eventos que no han sucedido pero podrían ocurrir. El análisis se ha realizado en una pequeña región de Dinamarca. Para validar los resultados de la GAN, se han analizado diferentes estadísticos horarios que caracterizan la intensidad, frecuencia y estructura espacio-temporal de las precipitaciones. Los resultados muestran como el modelo estudiado es capaz de simular de forma precisa la intensidad de la lluvia pero no es capaz de reproducir correctamente la frecuencia y la estructura espacio-temporal de la precipitación observada.

Palabras clave: GAN, desagregación temporal, precipitaciones, aprendizaje profundo, red neuronal

Abstract

In this project is studied the generation of hourly precipitation's maps from maps with a daily time resolution by using Generative Adversarial Networks (GANs). The objective is to analyze how good GANs are capable of generate hourly precipitation maps preserving the structure of the ones observed. This process is known as temporal disaggregation which is highly valuable for example in cases where only daily precipitation data is available or for explore events that have not yet occurred in order to improve the characterization of the rainfall. Analysis is performed over a small region of Denmark. To validate the results, several statistic are calculated in order to study how good hourly precipitation's maps generated are based on the intensity, frequency and space-time structure of them. Results show that intensity is well characterize but this model can not reproduce frequency nor space-time structure correctly.

Keywords: GAN, temporal disaggregation, precipitations, deep learning, neural networks

Índice

1. Introducción	4
2. Fundamentos teóricos	5
2.1. Desagregación temporal	5
2.1.1. Desagregación temporal usando GANs	6
2.2. Machine learning	7
2.2.1. Tipos de machine learning	7
2.2.2. Deep Learning	8
2.2.3. GAN	10
2.2.3.1. Aplicaciones	11
2.2.4. GANs Condicionales	11
2.2.5. WGANs	12
3. Análisis	13
3.1. Procedimiento experimental	13
3.1.1. Características de la GAN	14
3.2. Estadísticos para la validación de los resultados	15
3.3. Resultados	16
4. Conclusiones	26
5. Future work	27
6. Datos y códigos	28
7. Bibliografía	29

1. Introducción

En este proyecto se ha estudiado la desagregación subdiaria de las precipitaciones a partir del uso de una red generativa adversaria (GAN) basándose en el artículo (Scher, S. and Peßenteiner, S. 2021).

La desagregación temporal es de gran utilidad para el estudio de fenómenos de los que se precisa de una gran resolución temporal. Por ejemplo, el cambio climático impacta en las precipitaciones cambiando las características hidrológicas. Usualmente estos datasets tienen una resolución temporal baja que provoca que haya ciertos fenómenos que no se pueden estudiar. Este método que se va a utilizar podría dar una solución a estos problemas.

Los datos utilizados en el presente proyecto se descargan del instituto hidrológico y meteorológico sueco (SMHI), una autoridad experta perteneciente al ministerio de clima y empresa de Suecia [1]. Estos datos obtenidos consisten en información (precipitaciones) del radar que abarcan la zona de Suecia y Dinamarca y tienen una resolución temporal de 5 minutos. Se utilizan datos para el intervalo de tiempo (2009-2022). La zona a estudiar se reduce a una región danesa de 32x32km. El trabajo consiste en utilizar una GAN condicionada, previamente entrenada con datos horarios (2009-2016) para simular mapas de precipitaciones horarios introduciendo como condición las sumas diarias.

A la hora de validar los resultados, se tiene que estudiar las propiedades con las que hay que analizar las lluvias. Estas son su intensidad, la frecuencia y su estructura espacio-temporal. Para esto, se han analizado varios estadísticos para así comparar los datos observados con los simulados y de esta manera determinar como de buena es la generación con esta GAN.

El documento se divide en varias partes: primero se hace una introducción teórica tanto de la desagregación temporal como de las redes neuronales, en concreto de las GAN que es la que se utiliza en este proyecto. Después se analizan los resultados, presentando antes el procedimiento experimental que se ha seguido tanto para la generación de escenarios como para su validación. Por último, se presentan las conclusiones extraídas, así como posibles líneas de investigaciones futuras basadas en este proyecto.

2. Fundamentos teóricos

2.1. Desagregación temporal

La desagregación temporal consiste en a partir de series temporales de baja frecuencia obtener unas series temporales de una alta frecuencia y que conserve la estructura de las anteriores [2]. Con esto se hace referencia a que propiedades como por ejemplo la media y la suma de los datos deben de preservarse.

Este problema es bastante recurrente y no sólo en el campo de la climatología. Una de las principales razones por las que las series temporales suelen tener una resolución temporal baja es que es más fácil de almacenar ya que sino el tamaño de los datos aumentaría considerablemente. Esto puede afectar a muchos campos, por ejemplo, la economía, en la que se tienen series temporales de baja frecuencia, como pueden ser datos anuales o mensuales, y tal vez uno quisiera tener datos con una resolución temporal más alta, como puede ser semanal o diaria.

En cuanto a la climatología, encontrar colecciones con datos sub-diarios es muy difícil debido principalmente a que los instrumentos de medida automáticos se empezaron a instalar a principios del siglo XXI. Es por eso que es importante la desagregación temporal en la climatología. Y es que a través de datos con una frecuencia diaria se puede analizar el clima desde un punto de vista global pero muchos eventos meteorológicos extremos no se pueden observar con una resolución temporal diaria, por lo que sería necesario estudiarlo con una resolución temporal más alta. Estos eventos meteorológicos extremos son de gran interés porque tienen un gran impacto en la sociedad. Un ejemplo son las lluvias intensas en cortos períodos de tiempo que pueden provocar inundaciones, deslizamientos de tierra y erosión del suelo [5]. Entendiendo estos eventos extremos se pueden prevenir desastres elaborando planes de riesgos o teniéndolo en cuenta en el diseño de las infraestructuras. No sólo se puede aplicar a precipitaciones sino que se puede utilizar con otras variables meteorológicas como la temperatura. En el caso de las temperaturas, permite entender como varía la temperatura a lo largo del día. Con todos estos eventos extremos se puede estudiar mejor el impacto del cambio climático. También puede afectar a otros sectores como la agricultura o sectores energéticos (energía eólica o solar) [3].

Para realizar estas desagregaciones hay varias técnicas:

1. Métodos estadísticos que van desde simples interpolaciones hasta métodos no paramétricos o modelos que derivan relaciones estadísticas de datos históricos.
2. Generadores de clima, los cuáles generan secuencias sintéticas de las variables del clima horario a partir de generadores de números aleatorios que coinciden con los estadísticos [6].
3. Métodos mecanicistas: Los modelos climáticos regionales se utilizan para reducir dinámicamente las condiciones atmosféricas en el tiempo y el espacio [6].

En el caso concreto del clima, algunos ejemplos concretos que se han usado varios métodos para la desagregación temporal de la precipitación. Algunos de ellos son [29]

1. Los modelos de cascada multiplicativa, para el cuál un ejemplo puede ser (Müller and Haberlandt, 2018)
2. El método de fragmentos utilizado, entre otros, por (Sharma and Srikanthan, 2006)
3. El modelo Barlett-Lewis puesto en práctica en (Koutsoyiannis and Onof, 2001)
4. Redes neuronales artificiales como es el caso de (Kumar et al. 2012)

2.1.1. Desagregación temporal usando GANs

En este proyecto, se ha utilizado una red neuronal generativa adversaria para realizar la desagregación temporal, consiguiendo mapas de precipitaciones horarios a partir de los datos diarios. Para aplicar este modelo se necesita un histórico de datos con resolución temporal horaria para entrenar la red y luego datos diarios para introducirlos como condición y así obtener la desagregación temporal.

No es la primera vez que se propone esta solución al problema de conseguir datos con una resolución temporal más alta. De hecho, este proyecto se basa en el artículo (Scher, S. and Peßenteiner, S. 2021) [29]. En este artículo usan las GANs para que, a partir de los datos diarios, estimar la completa distribución de probabilidad de los patrones espaciales de precipitaciones para una resolución temporal horaria.

Los resultados que obtienen, a simple vista son indistinguibles de los datos reales pero analizando los distintos estadísticos realizados, observan que hay desviaciones. Por otro lado, señalan varias limitaciones que tiene su modelo. Estas son que no tiene en cuenta la topología del terreno, que tiene problemas en los bordes y que tal y como está diseñada la red, esta no es capaz de aprender la relación temporal y espacial de los datos [29].

2.2. Machine learning

El machine learning o aprendizaje automático es una rama de la inteligencia artificial (IA) que consiste en que la máquina aprenda de los datos, identificando patrones en vez de programarlo explícitamente [10], como se haría clásicamente. Esto está en auge debido a su gran utilidad y a los diversos campos que se pueden aplicar. Un esquema de la idea de machine learning se puede ver en la siguiente figura

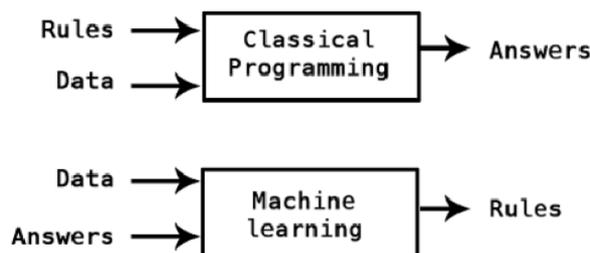


Figura 1: Esquema de proceso de machine learning frente a la programación clásica [32].

Se ve en la figura como para la programación clásica se introduce unos datos y unas reglas. Acorde a esas reglas y usando los datos introducidos la máquina nos devuelve las respuestas. El machine learning funciona de manera diferente. Se le introducen datos, y en función del tipo de machine learning, las respuestas y la máquina es la responsable de aprender las reglas a partir de esos datos.

2.2.1. Tipos de machine learning

Dentro del machine learning, en función de cómo se produce el aprendizaje, se pueden distinguir cuatro tipos de machine learning: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semi-supervisado y el aprendizaje por refuerzo [10].

1. El aprendizaje supervisado consiste en que en el proceso de entrenamiento se introducen los inputs con sus correspondientes outputs (datos etiquetados). De esta forma, una vez finalizado el entrenamiento, se le introduce un input y la máquina predice el output en base a lo aprendido en el entrenamiento. Dentro del aprendizaje supervisado se encuentran dos categorías: clasificación y regresión.
2. El aprendizaje no supervisado consiste en realizar el entrenamiento solamente con los inputs (datos no etiquetados). De esta forma la principal función de la máquina es encontrar patrones ocultos y así poder agrupar según las diferencias y similitudes que encuentre entre los datos. Se puede dividir en clustering y asociación.
3. El aprendizaje semi-supervisado, como su nombre indica, se basa en una mezcla entre los anteriores tipos de aprendizajes. En primer lugar, se entrena la máquina con unos pocos datos etiquetados (inputs and outputs) y después se aplica a una cantidad de datos más grandes si etiquetar [11].
4. El aprendizaje por refuerzo es un proceso basado en el feedback. Consiste básicamente en aprender de la experiencia. Cuando acierta recibe una recompensa y cuando falla

una penalización siendo el objetivo maximizar las recompensas. Se puede dividir en aprendizaje por refuerzo positivo y aprendizaje por refuerzo negativo [12].

En la siguiente figura se muestra un esquema de los distintos tipos de machine learning explicados anteriormente.

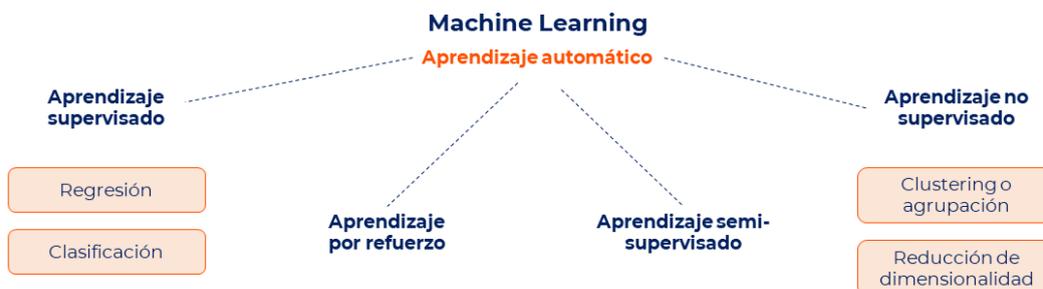


Figura 2: Esquema de los distintos tipos de machine learning en función de cómo es el proceso de aprendizaje [10].

2.2.2. Deep Learning

Dentro del machine learning se encuentra el deep learning o aprendizaje profundo. Sus fundamentos matemáticos ya existían desde el 1989 pero en la actualidad está en auge debido al avance en la computación que permite aplicar estas técnicas. En primer lugar, para el aprendizaje profundo es necesario una gran cantidad de datos (Big Data) y para ello se necesita tecnología capaz de almacenarlos, lo cual ha sufrido un avance durante ese intervalo de tiempo. Por otro lado, también requieren de un gran poder de computación, el cual se ha incrementado en los últimos años debido a la mejora de rendimiento de las CPU además del desarrollo de las GPUs.

El deep learning intenta imitar la forma en la que el ser humano aprende inspirándose en la neurociencia. Para imitar como aprende el ser humano, se utilizan redes neuronales en las que se basan principalmente estas técnicas de deep learning.

Una red neuronal consiste en un conjunto de procesadores o neuronas interconectadas entre sí que no están programadas rígidamente [15]. Estas neuronas se organizan en capas, las que se pueden dividir en capa de entrada, de salida y las capas ocultas. De aquí es de donde surge el concepto de aprendizaje profundo, que hace referencia a la existencia de estas capas (más capas, más profundo). Las neuronas tienen una función de activación que determinan como se combinan las entradas y se produce una salida [16]. Un ejemplo de red neuronal es la siguiente:

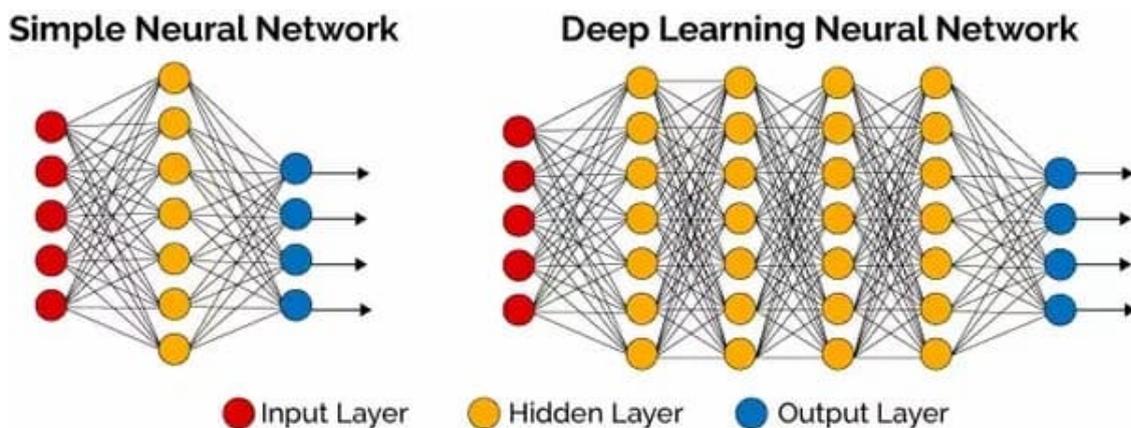


Figura 3: Se muestra un esquema de dos redes neuronales. A la izquierda una red neuronal con una sola capa oculta y a la derecha una red neuronal con más capas ocultas. En rojo se muestra las capas de entrada, en amarillo las capas ocultas y de azul las capas de salida [16].

En la figura se pueden observar dos redes neuronales, una simple con las capas de entrada y salida además de una capa oculta y otra red neuronal profunda con un mayor número de capas. El número de capas o neuronas es personalizable a la hora de construir una red neuronal.

Además del número de capas ocultas y de neuronas por capa, también hay que elegir las funciones de activación. Las funciones de activación más conocidos son ReLU (y Leaky ReLu), sigmoide, tanh y softmax. Por otro lado, para su entrenamiento hay que seleccionar una serie de parámetros como la función de pérdida (dependiendo de la tarea), el optimizador, la tasa de aprendizaje, el tamaño del batch y el número de épocas durante las que se quiere entrenar la red.

Dentro del deep learning se pueden distinguir varios tipos de redes neuronales:

1. Deep Neural Networks (DNN) o redes neuronales profundas son las que se han explicado previamente, básicamente consisten en una red neuronal con un mayor nivel de complejidad, es decir, un mayor número de capas ocultas (al menos 2).
2. Convolutional Neural Networks (CNN) o redes neuronales convolucionales que se utilizan principalmente para reconocer imágenes (topología de celda) [17]. Estas redes aprenden patrones locales e invariantes translaciones espaciales lo que hace que sean muy útiles para las imágenes. Además, puede aprender la jerarquía espacial de las imágenes donde la primera capa convolucional extrae bordes y esquinas mientras la segunda crea objetos más complejos. La arquitectura es distinta a las DNNs, introduciendo las capas de convolución entre otras. El final de la red consiste en una capa densa.
3. Recurrent Neural Networks (RNN) o redes neuronales recurrentes son aquellas redes neuronales que permiten que salidas obtenidas previamente se utilicen como entradas [18]. La principal característica de estas redes son el 'hidden state' que recuerda información de secuencias históricas [19]. Estas redes son utilizadas principalmente en reconocimiento de voz o en el campo de procesamiento del lenguaje natural.

4. Generative Adversarial Networks (GANs) o redes adversarias generativas que se explicaran en el siguiente apartado ya que son las que se han utilizado en este proyecto.

2.2.3. GAN

Una de las técnicas de Deep Learning que mejores resultados ofrecen son las de clasificación ya que los modelos generativos tuvieron menos impacto debido a sus limitaciones [21]. Estos modelos consisten en a partir de los datos de entrada crear nuevos datos indistinguibles de los reales. Un ejemplo de modelos generativos es el Latent Dirichlet Allocation (LDA) [20].

Las redes neuronales generativas adversarias (GANs) son modelos generativos basados en deep learning y consisten en la generación de nuevos datos a través de la competencia entre dos redes neuronales una que funciona como generador y la otra como discriminador [20].

1. Generador: Es la red neuronal que se encarga de generar nuevos datos a partir de ruido
2. Discriminador: Es la red neuronal que se encarga de distinguir si lo datos generados por el generador son distinguibles de los reales o no.

A continuación, se puede ver en la siguiente figura un esquema del proceso que siguen las redes neuronales adversarias.

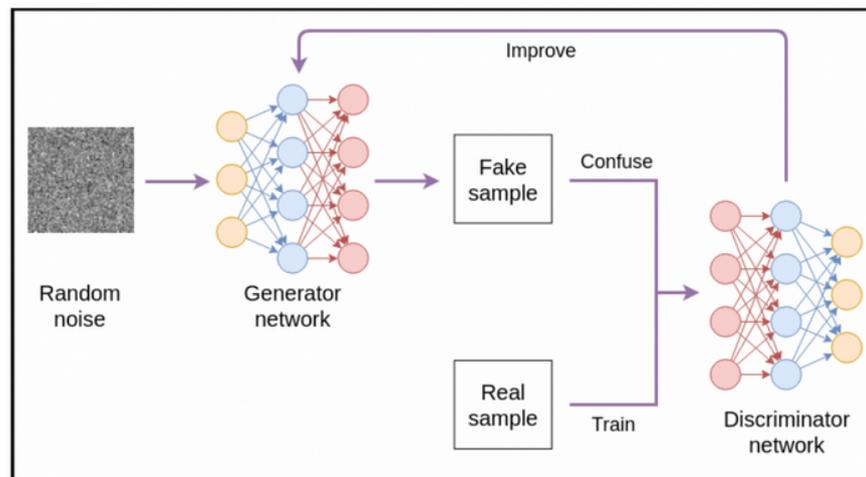


Figura 4: Esquema de funcionamiento de las redes generativas adversarias [30].

El proceso que sigue las redes neuronales generativas adversarias es el siguiente: Al generador se le pasa ruido y esta red genera unos datos que se van a llamar falsos. Por otro lado, se tiene el discriminador que recibe datos de entrenamiento reales y los datos falsos del generador. Este discriminador es un clasificador binario (supervisado porque los datos están etiquetados) que clasificara los datos en reales o falsos. Después con el feedback de este discriminador, el generador mejora la generación de los datos para así intentar engañar al discriminador y a su vez el discriminador también mejora [31].

A esto se le llama un juego de suma cero porque cuando uno gana el otro pierde y el agregado es 0. De esta forma, se observa una competición entre el generador y el discriminador donde el generador intenta engañar al discriminador mientras que este intenta distinguir las muestras

reales de las generadas y así se van mejorando ambos. Para esto, se utiliza una función de pérdida para ajustar los distintos parámetros [31]. El problema es que están involucradas dos redes neuronales donde cuando una gana la otra pierde, por lo que la función de pérdida no es sencilla. Para las redes neuronales generativas adversarias esta se define como

$$\min_G \max_D \mathcal{V}(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(D(x))] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

donde la D hace referencia al discriminador y G al generador. Consta de dos términos, el primero es el discriminador que intenta maximizar la función de pérdida y el otro término corresponde al generador que la intenta minimizar [22].

Como se ha visto en la explicación previa, las GANs no necesitan ni de inferencia aproximada ni de cadenas de Markov como modelos anteriores [21].

2.2.3.1. Aplicaciones

Aunque el término de redes neuronales generativas adversarias quizás no sea muy conocido, especialmente para el público general, tiene aplicaciones que son bastantes conocidas y que han tenido un gran impacto en la sociedad. Quizás la aplicación más conocida sea los deep-fakes, que son vídeos o imágenes generados de forma muy realistas para intentar manipular a la gente. Esta generación de contenido multimedia se lleva a cabo con GANs. Esta técnica de deep learning obtiene muy buenos resultados en tareas como por ejemplo, la generación de rostros realistas, los cuáles son muy difíciles de distinguir a simple vista.

2.2.4. GANs Condicionales

Las redes neuronales generativas adversarias condicionales son una extensión de las redes neuronales generativas adversarias. Estas siguen una arquitectura y procedimiento similar a las tradicionales, la diferencia es que ahora se añade una condición a los datos que genera el generador. En vez de pasar al generador ruido también se le proporciona una condición y lo mismo sucede con el discriminador que ahora recibirá los datos reales de entrenamiento, los datos falsos del generador y la condición [24]. De tal forma, que ahora el discriminador tiene que distinguir si los datos son reales o no y si además son coherentes con la condición que se le ha introducido. La idea del funcionamiento es la misma que las GANs, lo único que cambia es la función de pérdida [23] que se expresa como

$$\min_G \max_D \mathcal{V}(D, G) = \mathbb{E}_{(x,c) \sim p_{\text{data}}(x,c)}[\log(D(x|y))] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z|y)))] \quad (2)$$

La función de pérdida es muy parecida a las GANs tradicionales a diferencia de que se tiene en cuenta una condición que se introduce como input.

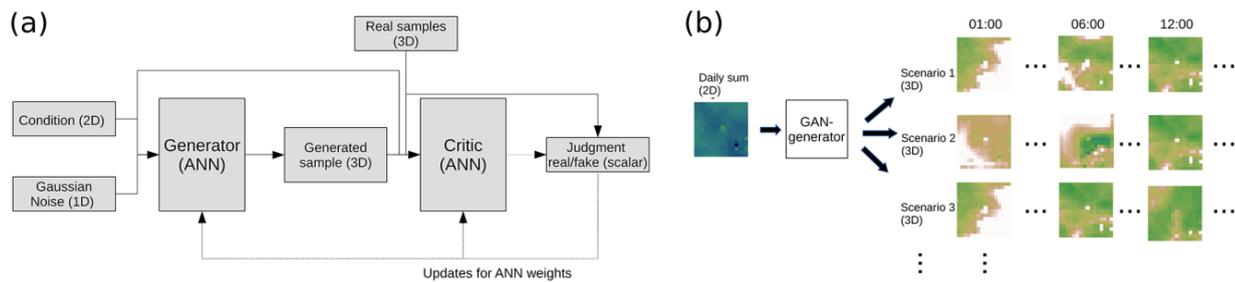


Figura 5: La imagen a) es un esquema de cómo funciona una red neuronal generativa adversaria mientras que la imagen b) es un ejemplo de cómo se generan los escenarios del proyecto [29].

En esta figura se puede observar en la imagen a) un esquema de cómo funcionan las GANs condicionales. En esa imagen lo que se llama 'Critic' es el discriminador. Tal y como se ha explicado antes, como al generador se le introduce ruido y la condición para que genere los datos mientras que al discriminador se le introduce esa misma condición junto a los datos reales y los generados. Esa red neuronal clasifica si son reales o no (siendo consistentes con la condición) y con ese feedback se actualizan los pesos de ambas redes neuronales.

En cuanto a la imagen b), se trata de un esquema de la generación de los distintos escenarios de precipitaciones que se pueden hacer con la GAN de este proyecto. Como condición se le introduce la suma diaria y esta red genera 10 escenarios distintos (se puede seleccionar otro número de escenarios) de mapas horarios de precipitaciones que sean consistentes con la condición introducida.

2.2.5. WGANs

Las Wasserstein Generative Adversarial Networks (WGANs) son una variante de las GANs tradicionales, propuesta en 2017. La principal diferencia entre ambos tipos de GANs es la distancia que se utiliza en la función de pérdida. Las WGANs utilizan la distancia Wasserstein en vez de utilizar la divergencia Kullback–Leibler o la divergencia Jensen–Shannon [26]. Con estas redes se aprecian dos ventajas principales.

1. Mejora de la estabilidad del proceso de entrenamiento y optimización del modelo, donde en el artículo original no se vio el modo colapso que a veces sucede en las GANs [25].
2. La función de pérdida, con la nueva distancia, es más significativa y está correlacionada con la calidad de la muestra y la convergencia del generador [25].

Una red neuronal generativa adversaria condicional de este tipo (WGANs) es la que se ha utilizado en el proyecto para generar las simulaciones de mapas de precipitaciones con resolución horaria a partir de muestras con una resolución diaria.

3. Análisis

3.1. Procedimiento experimental

Para realizar el análisis, se descargan los datos de radar que comprenden los datos en el intervalo de tiempo de 2009-2022 extraídos del Swedish Meteorological and Hydrological Institute (SMHI). El formato en el que se descarga la información es GeoTIFF y con una resolución temporal de 5 minutos. Una vez descargados todos los datos se transforman para que sean horarios y los guardamos en formato netcdf ya que es más fácil de manejar. El area de estudio comprende Suecia y Dinamarca, como se puede ver en la siguiente figura:

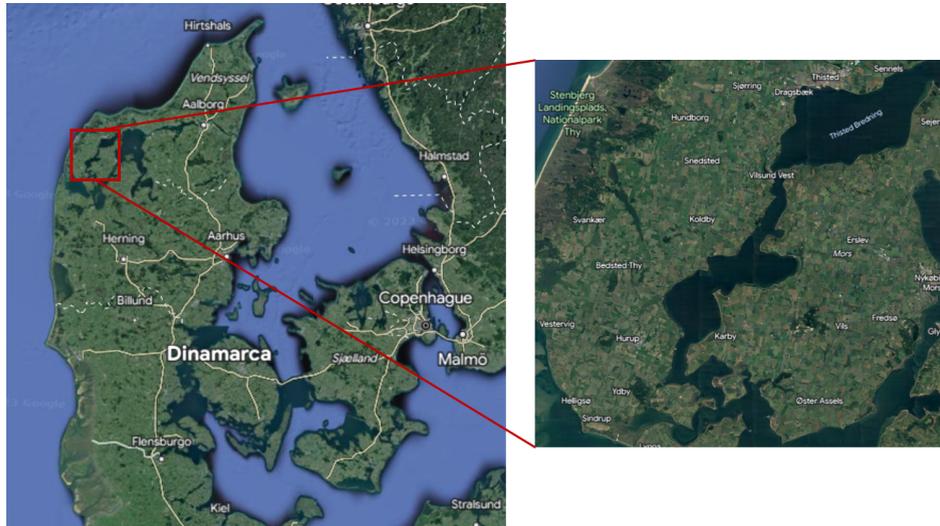


Figura 6: La imagen de la izquierda muestra el mapa de Dinamarca, el cual contiene la zona que se va a estudiar y la imagen de la derecha es dicha zona ampliada [27].

La imagen de la izquierda muestra Dinamarca y con un rectángulo rojo se ha marcado la zona que se va a estudiar. En la figura de la derecha se puede observar un zoom de la región de aproximadamente 32x32 km que se va a analizar. Es necesario reducir la zona de estudio, porque requiere mucho poder de computación analizar todo Suecia y Dinamarca durante un período largo de tiempo.

Para el análisis se va a utilizar una WGAN, que ya está preentrenada para el período de tiempo 2009-2016. Mediante este GAN preentrenado, se van a generar 10 escenarios por día introduciendo datos correspondientes al intervalo de tiempo 2017-2022 que se utilizarán para la validación. Como se trata de una GAN condicionada, hay que introducir la condición que es la suma diaria.

También se lleva a cabo otro filtrado y es que se van a considerar sólo los días en los que llueve y por lo tanto se eliminan los días que en esa región seleccionada la lluvia sea menor de 5 mm/día y por supuesto los días que tengan valores NaN.

3.1.1. Características de la GAN

En este proyecto se ha utilizada una WGAN con pixel normalization y penalización de gradiente [29]. Como se ha explicado anteriormente, las GAN tienen un generador y un discriminador.

La arquitectura del generador es la siguiente:

1. El input del generador es un vector de números aleatorios de una longitud 100 y un vector unidimensional con la condición de la suma diaria [29].
2. Una capa densa de tamaño $256 \times 2 \times 2 \times 3$ con una función de activación Leaky ReLU ($\alpha=0.2$). Después se aplica un Reshape para cambiar las dimensiones de $(256 \times 2 \times 2 \times 3)$ a $(3, 2, 2, 256)$.
3. Se aplica un remuestreo 3D con la capa UpSampling3D con un tamaño $(2,2,2)$, una capa de convolución de 256 filtros (kernel size = $(3,3,3)$ y padding='same'), la función de activación Leaky ReLU ($\alpha=0.2$) y pixel normalization.
4. Se repite dos veces el paso anterior pero las capas de convolución tienen 128 y 64 filtros en cada paso respectivamente.
5. Capa de convolución de un solo filtro (kernel size = $(3,3,3)$) con una activación lineal.
6. Función de activación Softmax sobre el eje de las horas. Se trata de una regresión logística sobre esta dimensión [29].

Por otro lado, se tiene el discriminador cuya arquitectura consiste en:

1. La entrada que es un tensor 3D de dimensiones $(nhours, 16, 16, 2)$
2. 4 capas de convolución cuyo número de filtros va aumentando $(64, 125, 256, 256)$ y strides=2 con funciones de activación Leaky ReLU ($\alpha=0.2$) y dropout(0.25)
3. Se aplica una capa Flatten() para posteriormente añadir la capa final densa de tamaño 1 y con una función de activación lineal.

Ambas redes se entrenan para 50 épocas y se utiliza el optimizador Adam.

3.2. Estadísticos para la validación de los resultados

Una vez que se han generado los distintos escenarios para los distintos periodos de tiempo, hay que validar estos resultados. Para esto, se han utilizado varios estadísticos. Y es que, a la hora de estudiar las precipitaciones, no solo es importante conocer la cantidad de agua que llueve. Se tiene que tener en cuenta tres características principales: la intensidad, la frecuencia y la estructura espacio-temporal de las mismas.

Los estadísticos utilizados han sido la media, la varianza, el skewness, la probabilidad de un periodo seco ϕ^h , la probabilidad de dos periodos consecutivos con lluvia ϕ^{WW} y la probabilidad de dos periodos consecutivos secos ϕ^{DD} . Además de la correlación espacial y autocorrelación temporal con lag de 1 hora. La intensidad se caracteriza con la media, la varianza y momento de tercer orden (skewness) donde este último se puede utilizar para caracterizar los extremos. La frecuencia se caracteriza mediante la probabilidad de un período seco (o de uno lluvioso) ϕ^h . Por último, el resto de estadísticos, correlación espacial, la autocorrelación temporal y las probabilidades de transición, caracterizan la estructura espacio-temporal de las precipitaciones.

Para comparar los datos observados con los datos simulados los estadísticos se aplican de la siguiente manera:

1. Se tienen datos divididos para cada estación (verano, primavera, otoño, invierno) ya que dependiendo del clima puede haber grandes diferencias en las precipitaciones entre las estaciones por lo que se estudiarán por separado (clima no estacionario).
2. Por cada estación del año se tienen los datos observados extraídos directamente del SMHI y transformados de la forma explicada anteriormente
3. En cuanto a los datos simulados, los estadísticos se aplican a uno de los escenarios generados (se le asigna el nombre de datos simulados), a los valores máximos de los 10 escenarios generados (Max (10 sim)) y a los valores mínimos de los 10 escenarios (Min (10 sim)).

De esta forma, se puede comparar los datos observados y simulados y al representar los casos máximos y mínimos de cada simulación se puede analizar su variabilidad.

3.3. Resultados

En primer lugar, se estudia como son las lluvias a lo largo del año en la zona que se quiere estudiar. Para ello, se han obtenido las precipitaciones mensuales de una población que se encuentra dentro de esa región que se está estudiando.

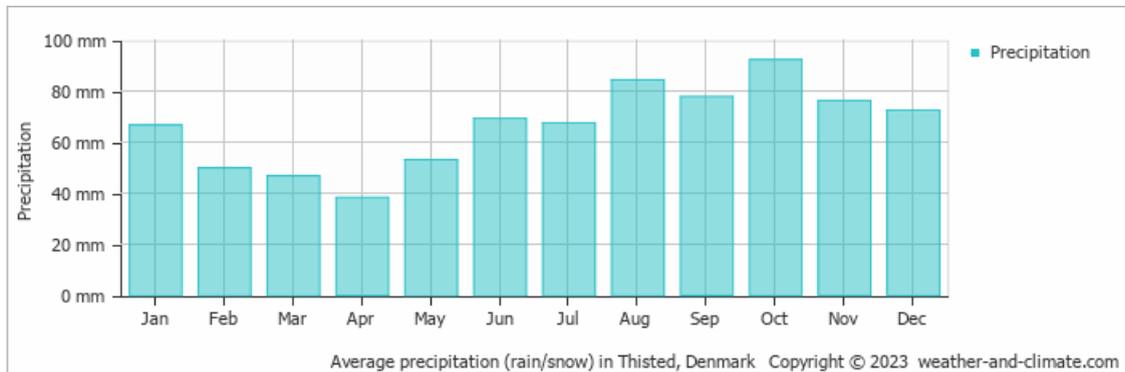


Figura 7: Precipitaciones mensuales de la región de Thisted en mm [28].

De esta forma, se puede comparar con los datos que se han extraído tanto los observados como los que se han generado. Se observa claramente como las estaciones más lluviosas son verano y otoño mientras que primavera es la estación del año más seca.

Ahora se va a mostrar un ejemplo de la generación de escenarios horarios.

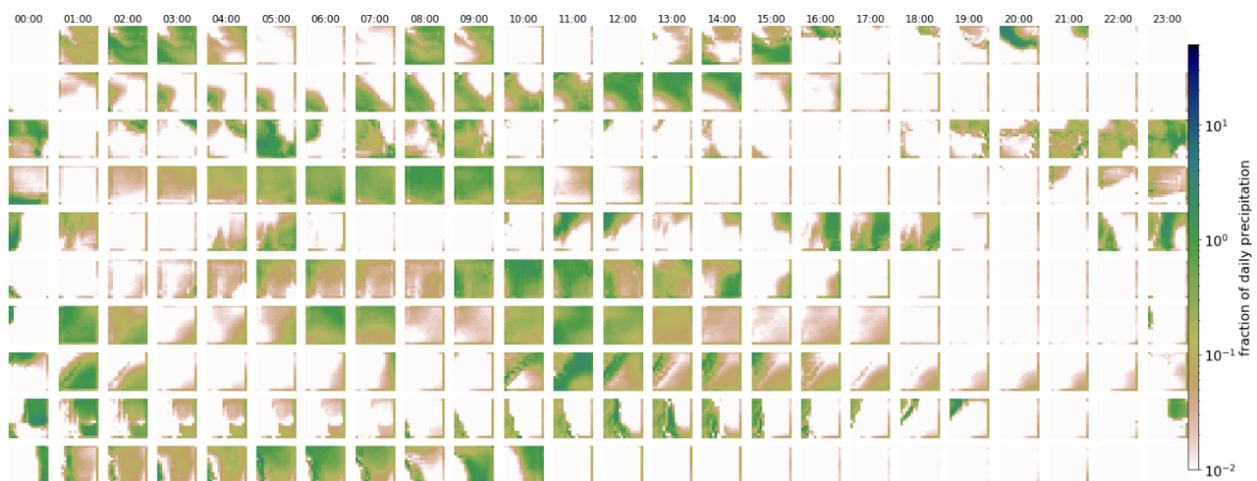


Figura 8: Ejemplo de escenario simulados para un día concreto donde se representa la fracción de la lluvia diaria.

En esta figura se observa la fracción de lluvia diaria para las 24 horas y para un total de 10 escenarios generados. Este sería el output obtenido al introducir como condición el primer día de verano de 2017.

Después, se muestra la probabilidad de excedencia para las cuatro estaciones del año, donde se comparan los datos observados frente a la simulación.

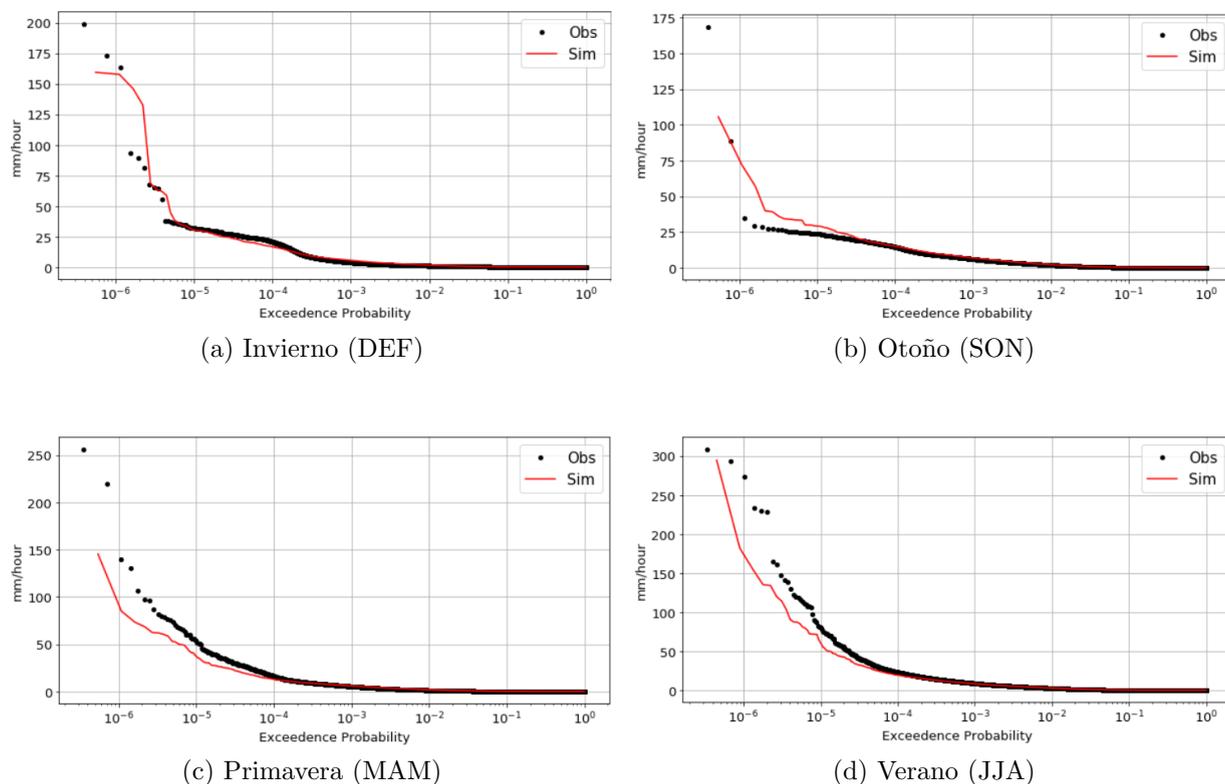


Figura 9: Probabilidades de excedencia para las cuatro estaciones del año, donde los datos observados son los puntos negros y la línea roja los simulados. Arriba a la izquierda es el invierno, arriba a la derecha es el otoño, abajo a la izquierda corresponde con la primavera y abajo a la derecha el verano.

Se puede observar en la figura anterior como existe una buena correspondencia de la simulación respecto a los datos observados en cada estación. Para una alta probabilidad de excedencia los datos simulados coinciden de forma muy precisa con los datos observados mientras que para una probabilidad de excedencia inferior a 10^{-5} esa relación no es tan precisa.

Por un lado, se tiene el caso del invierno (arriba a la izquierda), que parece ser la estación con una correlación más precisa entre datos simulados y observados. Caso opuesto es el otoño (arriba a la derecha) para el cual hay una buena concordancia entre datos simulados y observados excepto alrededor de 10^{-6} donde se produce un cambio brusco en los datos observados que la simulación no es capaz de predecir. Después están las estaciones de verano y primavera (parte inferior), los cuáles son muy similares, con una buena precisión hasta 10^{-5} donde a pesar de seguir la misma tendencia, los datos simulados son algo inferiores a los observados.

A continuación, para validar los resultados de la GAN, se han analizado distintos estadísticos. Estos estadísticos se calculan para cada estación del año con los datos observados, los datos simulados en uno de los escenarios, los valores máximos de las 10 simulaciones y los valores mínimos de las 10 simulaciones.

En primer lugar, se ha calculado la media, lo que nos permite ver como en los distintos casos se distribuye la intensidad de la lluvia.

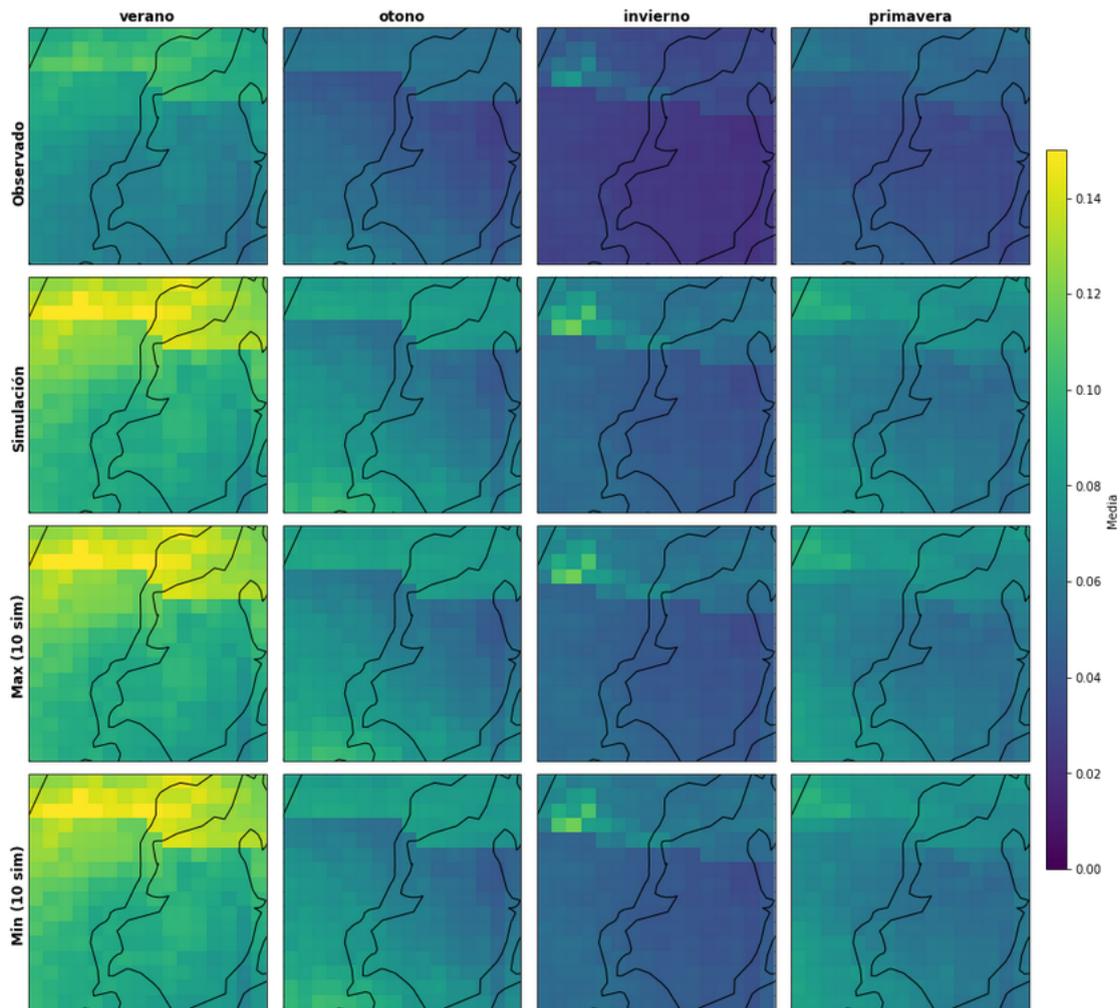


Figura 10: Media de las precipitaciones para cada estación del año sobre los datos observados, sobre una simulación y con los valores máximos y mínimos de las 10 simulaciones.

Observando los resultados de la media, se puede ver (cómo es lógico al tratarse de una media) que tanto los resultados para la muestra simulada, como para los valores máximos y mínimos son los mismos. Los datos observados y los simulados de cada estación deberían ser iguales, pero se aprecia como sistemáticamente los valores de los datos simulados son ligeramente superiores a los observados. Las zonas cuyos valores son más altos se conservan en ambos casos (observación y simulación). Y es que en las 4 estaciones se puede observar cómo hay una zona, en la parte superior de la región, que es mayor que el resto y esto coincide tanto

para los datos observados como los simulados. También si se presta atención a los datos del invierno, en la zona superior izquierda hay unos valores atípicos que aparecen tanto en los datos observados como en los simulados.

A continuación, se analiza la varianza para ver cómo están dispersos los datos de la media en las observaciones y compararlo con los simulados.

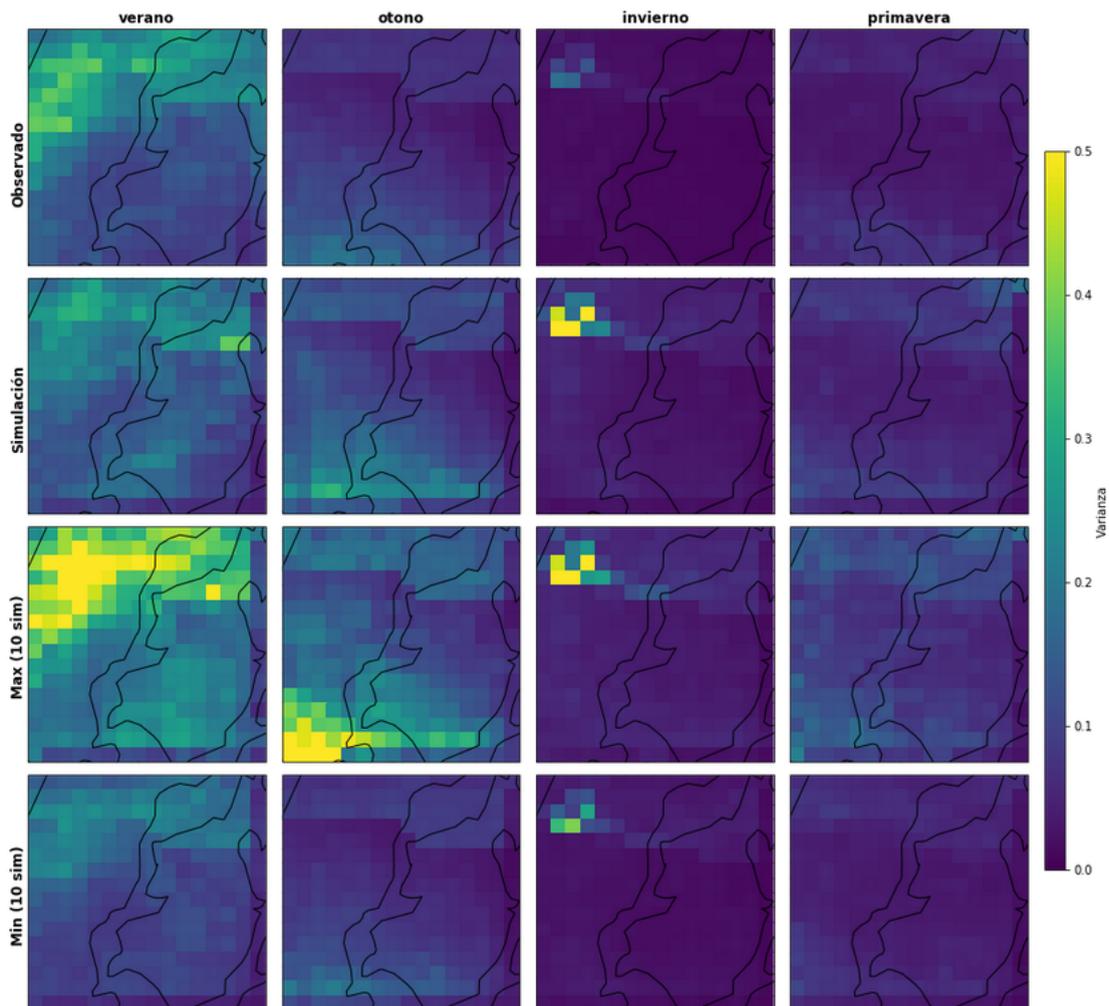


Figura 11: Varianza de las precipitaciones para cada estación del año sobre los datos observados, sobre una simulación y con los valores máximos y mínimos de las 10 simulaciones.

En este caso, también se pueden ver similitudes entre los datos observados y los simulados. Por ejemplo, en verano se ve una zona con valores más altos, en la zona superior e izquierda, que queda más patente mirando los valores máximos. Parecido sucede en otoño, pero en la parte inferior, donde hay un patrón algo más sutil que aparece tanto en los datos observados como en los simulados, siendo el mínimo de las 10 simulaciones muy similar a los datos observados. En invierno se conserva la zona de valores más altos que ya se apreciaba en la media. Para las varianzas, los valores entre los datos observados y los simulados son más parecidos que en el caso de la media.

Con el fin de estudiar la asimetría de los datos y caracterizar los extremos, se ha representado el skewness (momento de tercer orden) de la misma forma que los anteriores estadísticos.

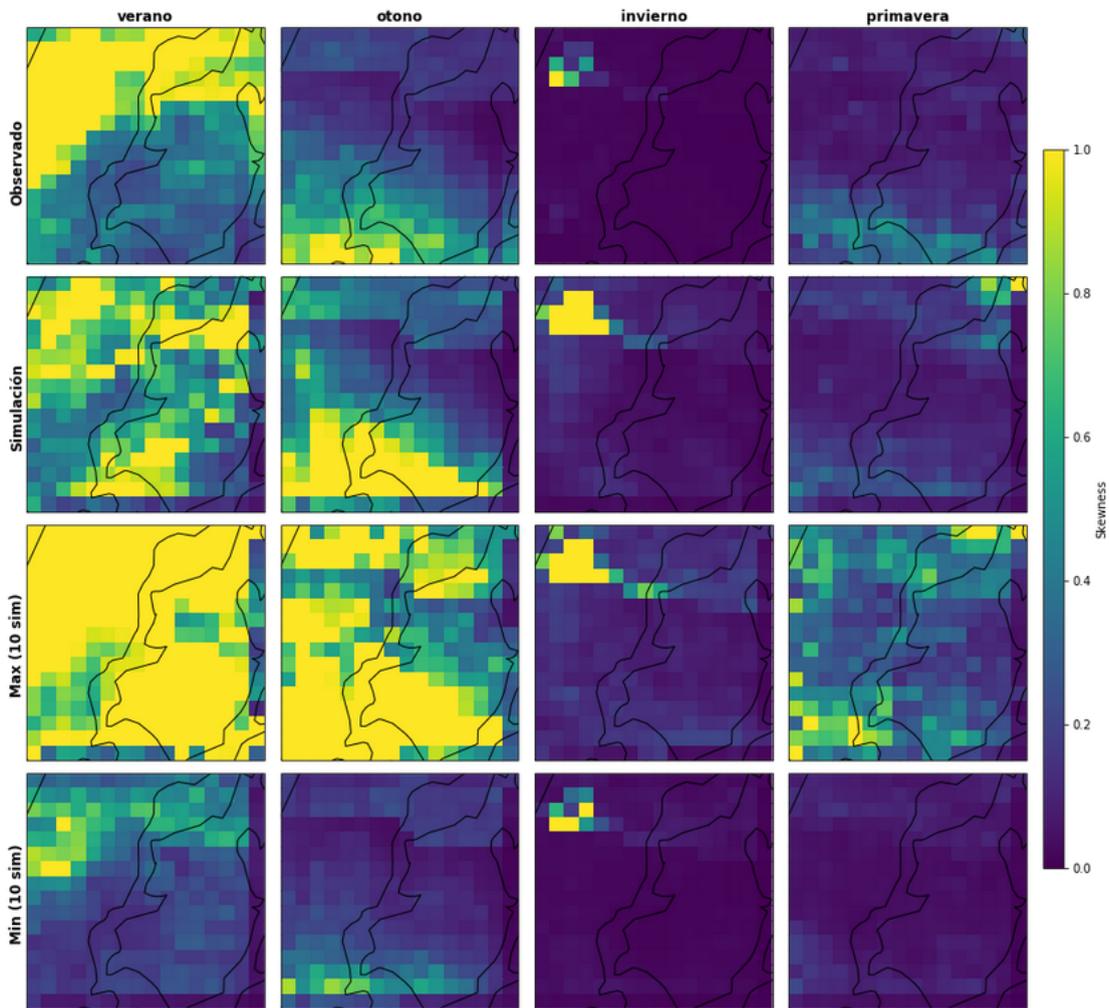


Figura 12: Skewness de las precipitaciones para cada estación del año sobre los datos observados, sobre una simulación y con los valores máximos y mínimos de las 10 simulaciones.

Observando los resultados del skewness, se puede ver que los resultados ya no son tan precisos como en los casos anteriores. A excepción del invierno donde en el caso de los valores mínimos de las 10 simulaciones es muy similar a los datos observados, además de que la simulación y los valores máximos tienen esa zona característica de esa estación. En el resto de los casos, parece que los patrones los conserva pero de forma mucho menos precisa, en las simulaciones aparecen valores altos donde en los datos observados no se aprecia demasiado. Además, los valores máximos son muy extremos y oscilan respecto a los observados.

Una vez analizado la media, varianza y el skewness se han calculado ϕ , ϕ^{WW} y ϕ^{DD} que son la probabilidad de un periodo seco, la probabilidad de dos periodos de tiempo consecutivo lloviendo y la probabilidad de dos periodos consecutivos de tiempo secos respectivamente.

En primer lugar, se ha calculado la probabilidad de un período seco.

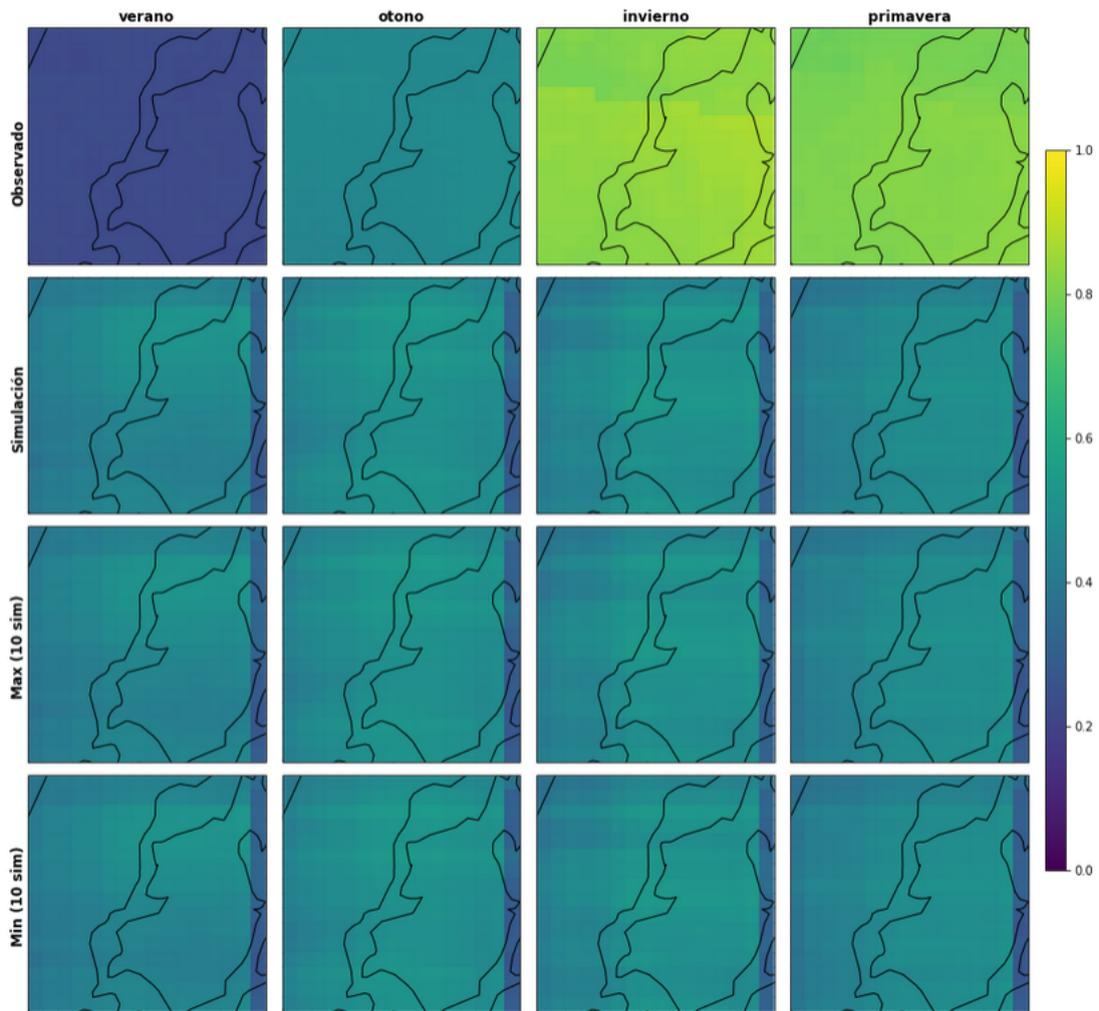


Figura 13: Probabilidad de un periodo seco para cada estación del año sobre los datos observados, sobre una simulación y con los valores máximos y mínimos de las 10 simulaciones.

En esta figura se observa la probabilidad de un período seco y por lo tanto los valores van de 0 a 1. Observando las figuras simuladas se puede apreciar que hay problemas de contorno en los bordes (también se apreciaba ligeramente al analizar el skewness). Comparando datos observados y datos simulados, se ve que solo coinciden en otoño y que la diferencia entre valores máximos y mínimos es muy pequeña. Por otro lado, los resultados obtenidos para los datos observados coinciden con lo esperado porque verano y otoño son las estaciones más lluviosas (cómo se ha visto en las precipitaciones mensuales) y por lo tanto una menor probabilidad de un período seco. Lo contrario ocurre con invierno y primavera que son estaciones más secas y por lo tanto habrá una mayor probabilidad de períodos secos.

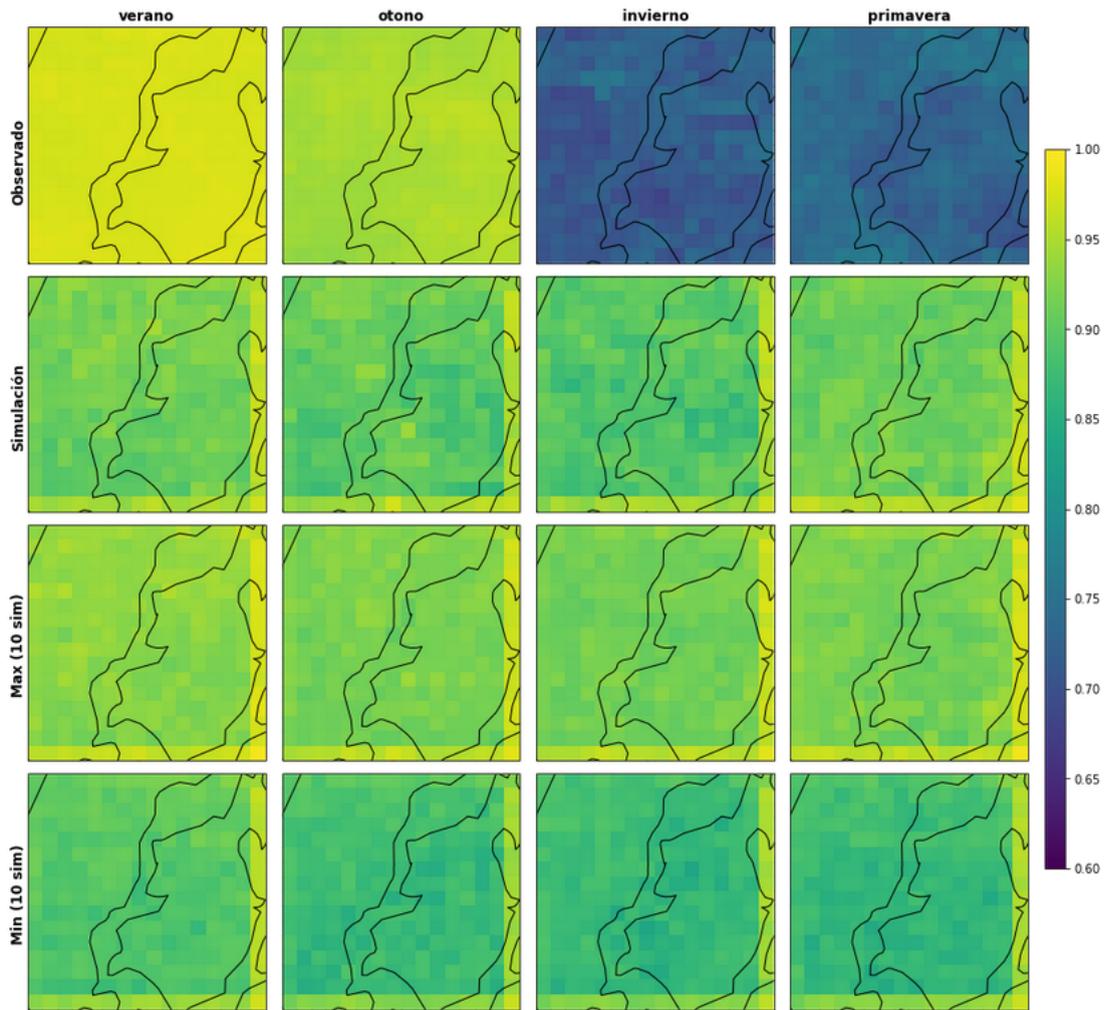


Figura 14: Probabilidad de dos periodos de tiempo consecutivo lluviosos para cada estación del año sobre los datos observados, sobre una simulación y con los valores máximos y mínimos de las 10 simulaciones.

Al analizar la probabilidad de dos periodos de tiempo consecutivo lluviosos, se siguen observando los problemas de borde. En cuanto a los datos observados, ocurre lo opuesto al caso anterior, lo que era de esperar. La probabilidad de que llueva en dos tiempos consecutivos será mayor en las estaciones que más llueva (verano y otoño) y menor en las que menos llueva (primavera e invierno). En cuanto a los datos simulados ocurre lo mismo que en el caso anterior, los valores no oscilan mucho en los casos extremos (máximos y mínimos) y todas simulaciones independientemente de la estación son muy parecidos. Se puede observar que los datos no son capaces de reproducir el comportamiento no estacional de los datos observados.

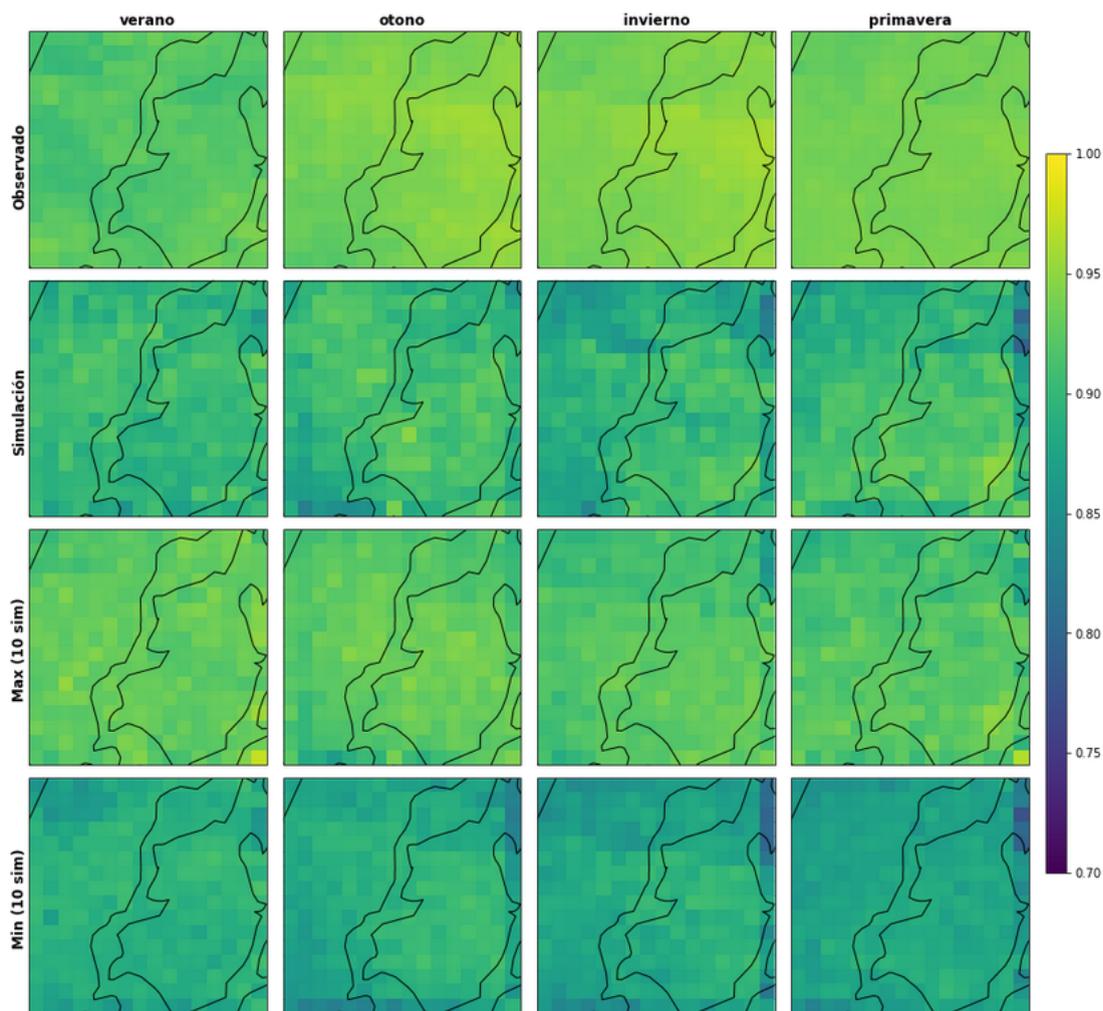


Figura 15: Probabilidad de dos periodos de tiempo consecutivos secos para cada estación del año sobre los datos observados, sobre una simulación y con los valores máximos y mínimos de las 10 simulaciones.

En el caso de la probabilidad de dos tiempos consecutivos secos, se esperaría lo contrario que en el estadístico anterior y se puede observar que la tendencia parece ser esa pero las diferencias son mucho menores que en el caso anterior. La estación del año que más se diferencia del resto es el verano donde los valores son menores, lo esperado ya que en esta estación las lluvias son más frecuentes. También en esta estación se ve una similitud entre los datos observados y simulados.

Una vez representadas las tres probabilidades, se va a analizar cómo responde los datos generados a la autocorrelación temporal con lag=1h.

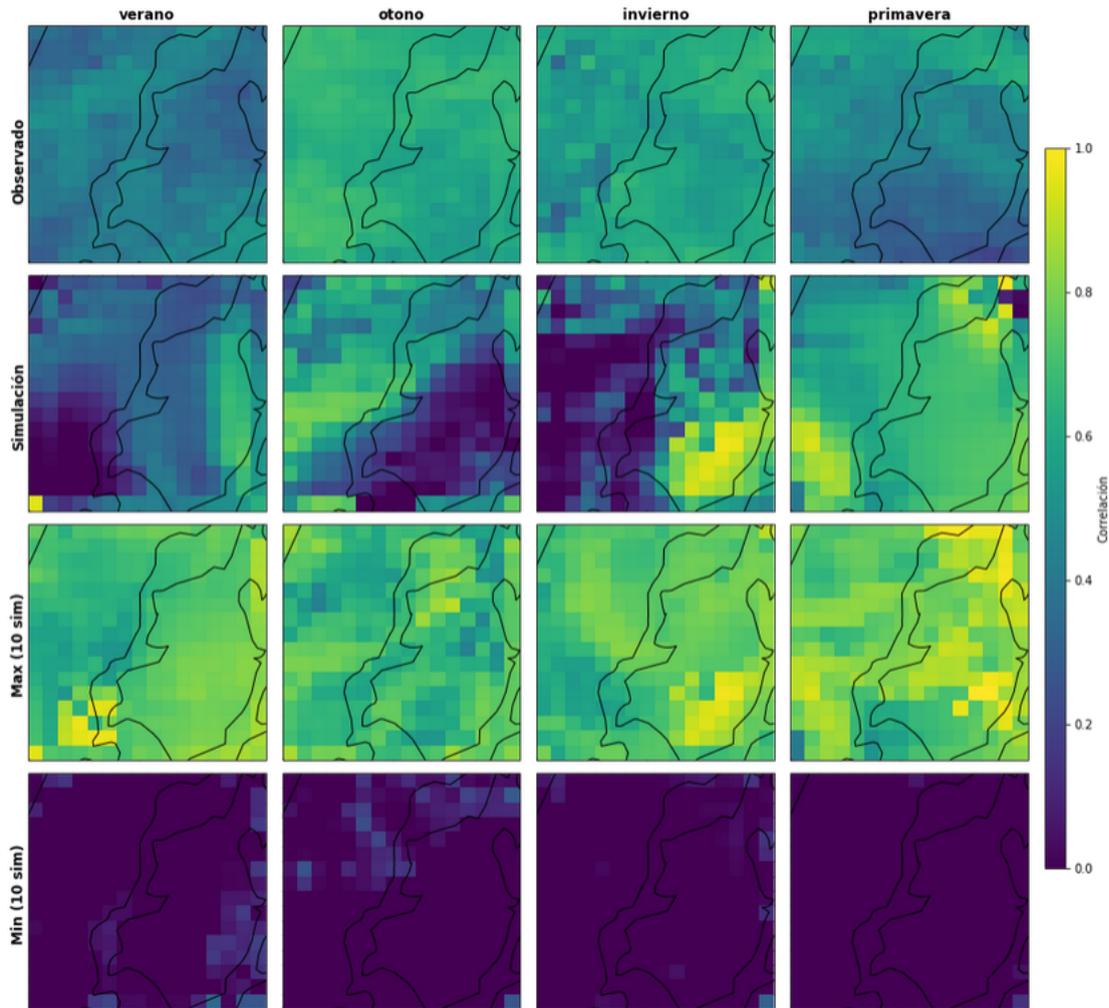


Figura 16: Autocorrelación temporal para cada estación del año sobre los datos observados, sobre una simulación y con los valores máximos y mínimos de las 10 simulaciones.

Este estadístico deja de manifiesto una serie de aspectos muy interesantes. Por un lado, los datos observados parecen ser bastante homogéneos sin ninguna zona con valores más altos o bajos respecto al resto. Entre los meses tampoco hay grandes diferencias, verano y primavera parecen tener una autocorrelación temporal algo menor que otoño e invierno. En cuanto a los datos simulados, se puede apreciar que no se parecen en nada a los datos observados. Fijándose en la simulación, se aprecia que varía mucho en función de la zona, justo lo contrario a los datos observados. Además, hay una gran diferencia entre los valores mínimos (prácticamente todo 0) respecto a los máximos (valores muy altos). Todo esto, indica que los datos simulados no reproducen bien la autocorrelación temporal, cuyos resultados parecen bastante aleatorios.

Por último, se va a estudiar la correlación espacial de los datos observados y de los datos simulados. La muestra está dividida en 16 puntos, por lo que para calcular la correlación espacial se calcula la distancia entre cada par de puntos y su correlación de Pearson. Como para una misma distancia hay varias correlaciones distintas, se hace la media de las corre-

laciones para cada distancia y se calcula su desviación estándar para así medir como de dispersos están los datos.

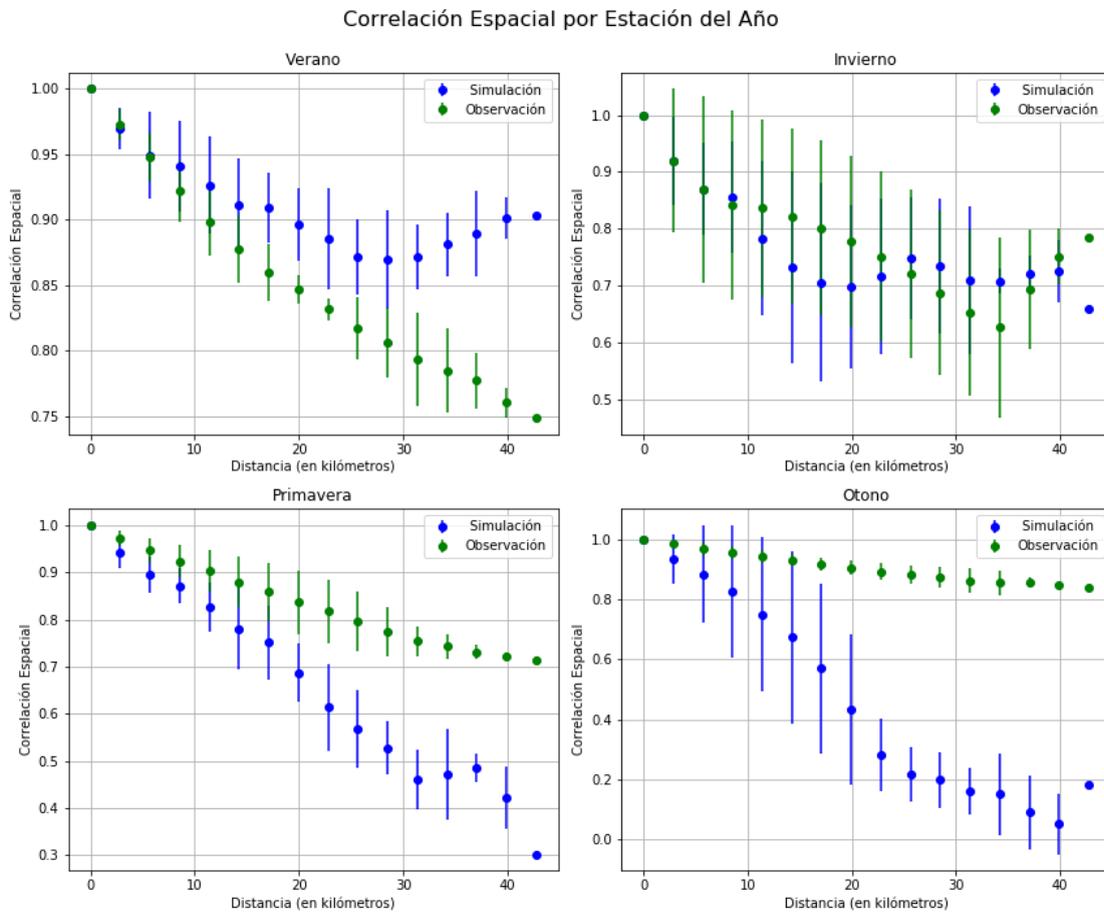


Figura 17: Correlación espacial para cada estación donde los puntos verdes son los datos observados y los azules son los simulados.

Analizando los resultados se puede ver como en las estaciones de primavera y otoño, tanto los datos simulados como los observados tienen una tendencia descendente con la distancia (lo cual es lo esperado) y los datos observados tienen una correlación espacial mayor en comparación con los simulados. La diferencia entre ellos es significativamente mayor a partir de los 20 km de distancia. El caso del verano es diferente ya que la correlación espacial de los datos observados y simulados son bastante similares hasta los 30 km de distancia donde los datos observados están dentro de las desviaciones estándar de los simulados y por lo tanto son compatibles. A partir de los 30 km de distancia los datos simulados no siguen la trayectoria descendente esperada, como si lo hacen los datos observados. Por último, en invierno se observan que las correlaciones para esta estación son similares, siendo los datos observados y simulados compatibles.

4. Conclusiones

En este proyecto se utiliza un WGAN basada en el artículo (Scher, S. and Peßenteiner, S. 2021) en el cual se estudia la desagregación temporal usando esta red generativa adversaria. La red utilizada está pre-entrenada con datos de Suecia y Dinamarca del intervalo de tiempo 2009-2016 y para generar los resultados se ha utilizado como condición la suma diaria del periodo 2017-2022 para una pequeña región de Dinamarca (considerando sólo días lluviosos).

Para la validación de la GAN se han utilizado distintos estadísticos, de tal forma que se puede estudiar una gran variedad de información de las lluvias. Esto incluye además de la intensidad de las lluvias, la frecuencia y la estructura espacio-temporal de las mismas. Estos estadísticos son: la media, la varianza, el skewness, las probabilidades ϕ , ϕ^{WW} y ϕ^{DD} y las correlaciones espaciales y temporales. También se ha estudiado la probabilidad de excedencia.

Los resultados obtenidos para la media, varianza y skewness son medidas de la intensidad de las precipitaciones. Estos estadísticos muestran que la GAN es capaz de generar escenarios cuya intensidad sea acorde a los datos observados. Se aprecia, tanto para la varianza como para la media, como los patrones (zonas con valores más altos o bajos respecto al resto de la región) coinciden para los datos observados y los simulados, incluyendo los casos de los máximos y mínimos de las 10 simulaciones. En el caso del skewness, que permite caracterizar los extremos, estos patrones no son tan claros entre los datos simulados y observados ya que los valores máximos oscilan mucho.

El caso de la frecuencia de las lluvias, se evalúa su rendimiento con el estadístico que mide la probabilidad de un período de tiempo seco. Para este estadístico se ha observado que las simulaciones no son acordes a los datos observados salvo en el caso del otoño.

Para las dos probabilidades de transición, se puede ver que las probabilidades cuadran con los datos observados porque en las estaciones en las que más llueven hay más probabilidad de periodos de tiempo lluviosos y viceversa. En cuanto a la relación con los datos simulados, aparecen varios problemas donde se pueden ver las carencias que tiene este modelo. En primer lugar, la existencia de problemas de borde, especialmente en la parte derecha e inferior. Y después, a pesar de que en las simulaciones no hay datos extremos, las situaciones máximas y mínimas tampoco oscilan mucho, pero no coinciden con los datos observados (salvo en verano para la probabilidad de dos intervalos secos).

En el caso de la autocorrelación temporal, no hay ninguna relación entre los datos observados y los simulados. De hecho, los resultados generados en la simulación son bastantes aleatorios, donde las situaciones máximas y mínimas son casos extremos mientras que en la simulación aparecen patrones extraños que no se aprecian en los datos observados. Esto es lo esperado ya que la GAN puede predecir bien la intensidad de la lluvia, pero a esta red neuronal en ninguno caso se le aplica ningún constraint para que pueda aprender la relación temporal de los datos por lo que se puede concluir que la autocorrelación temporal no es capaz de simularla correctamente.

En cuanto a la correlación espacial, los resultados son poco precisos al igual que para la autocorrelación temporal. Se puede observar que en invierno las correlaciones de los datos observados y simulados son compatibles para todas las distancias. Por otro lado, en las estaciones de primavera y otoño, a partir de los 20km, las correlaciones difieren mucho

entre los datos observados y los simulados, mientras que en verano las correlaciones son similares hasta los 30km. Por todo esto, no parece que la GAN sea capaz de generar escenarios conservando la correlación espacial, especialmente en el caso de distancias grandes.

Tras analizar todos los resultados se concluye que este método de desagregación temporal es capaz de generar escenarios precisos en cuanto a la intensidad de la lluvia pero los resultados no son buenos para la frecuencia y especialmente inadecuados para caracterizar la estructura espacio-temporal.

5. Future work

Una vez finalizado el proyecto, se pueden plantear futuras mejoras o ampliar el proyecto en otras zonas. En cuanto a las mejoras, a lo largo de este proyecto se han visto los puntos fuertes y débiles de este método. Concretamente se ha visto como la intensidad concuerda muy bien con los escenarios simulados, en cambio no sucede lo mismo con la frecuencia y la estructura espacio-temporal. Hay varias soluciones que se pueden plantear para mejorar estos resultados:

1. Ajustar 4 modelos, uno para cada estación del año. De esta forma se resolvería el problema de la no estacionariedad de los datos.
2. Analizar el uso de Redes Neuronales Recurrentes (RNR) para poder caracterizar la estructura temporal de los datos.
3. Añadir a la GAN constraints espaciales, relacionados con la topología del terreno, para tener en cuenta por ejemplo las cadenas montañosas, que pueden afectar a las precipitaciones de esa zona en particular.

Por otro lado, una posibilidad para aplicar este análisis puede ser utilizar esta GAN para datos de la península ibérica. Se podría plantear el uso de los datos de ERA5 [34] que forman parte del proyecto Centro Europeo de Predicción a Plazo Medio del Tiempo (ECMWF) [33]. Se tendría un intervalo mayor de tiempo (1978-2023) con lo que habría entrenar la GAN con estos datos. También sería necesario adaptar la GAN a estos nuevos datos e incluso se podría plantear la modificación de algunas de las capas para intentar obtener mejores resultados.

6. Datos y códigos

Este proyecto ha seguido los principios FAIR (Findability, Accessibility, Interoperability and Reuse), que en español se puede traducir como que los datos son localizables, que son accesibles, interoperables y que se pueden reutilizar. Los datos de radar de este proyecto se han descargado mediante la API del Instituto hidrológico y meteorológico sueco (SMHI). El proyecto se ha basado en el artículo (Scher, S. and Peßenteiner, S. 2021), el cual tiene en un repositorio github (<https://github.com/sipposip/pr-disagg-radar-gan/tree/master>) los scripts y demás información utilizada en el proyecto. De este repositorio, se ha utilizado concretamente los modelos entrenados, así como el script para cargar dichos modelos. El análisis no se ha hecho en local, debido a que algunos procesos necesitaban de muchos recursos computacionales, por lo que los scripts se han ejecutado en un jupyterhub del IFCA. Después se ha creado un repositorio github (https://github.com/pablo23hf/TFM_datascience) dónde se almacenan tanto los scripts utilizados en este trabajo fin de máster, bien sean para la transformación de los datos como para la obtención de los resultados, así como los datos utilizados (ya procesados) y el environment en el que se ha trabajado (archivo .yml).

7. Bibliografía

- [1] Forsgard, J. (2023). *Who we are*. SMHI. <https://www.smhi.se/en/about-smhi/who-we-are/who-we-are-1.83748>
- [2] Chamberlin, G. (2010). *Methods explained: Temporal disaggregation*. *Economic & Labour Market Review*, 4, 106-121.
- [3] Zabel, F., & Poschlod, B. (2023). *The Teddy-Tool v1. 0: temporal disaggregation of daily climate model data for climate impact analysis*. *Geoscientific Model Development Discussions*, 1-19.
- [4] Sax, C., & Steiner, P. (2013). *Temporal disaggregation of time series*.
- [5] Pritchard, D., Lewis, E., Blenkinsop, S., Patino Velasquez, L., Whitford, A., & Fowler, H. J. (2023). *An Observation-Based Dataset of Global Sub-Daily Precipitation Indices (GSDR-I)*. *Scientific Data*, 10(1), 393.
- [6] Förster, K., Hanzer, F., Winter, B., Marke, T., & Strasser, U. (2016). *An open-source MEteoroLOGical observation time series DISaggregation Tool (MELODIST v0. 1.1)*. *Geoscientific Model Development*, 9(7), 2315-2333.
- [7] Jesús Peñil, M. D., & Díez Sierra, J. (2023). *Climate change effects on sub-daily precipitation in Spain*.
- [8] Díez-Sierra, J., Navas, S., & del Jesus, M. (2023). *NEOPRENE v1. 0.1: A Python library for generating spatial rainfall based on the Neyman-Scott process*. *EGUsphere*, 2023, 1-19.
- [9] Chen, B. (2007). *An empirical comparison of methods for temporal distribution and interpolation at the national accounts*. BEA.
- [10] decide4AI. (2022, 16 noviembre). Tipos de aprendizaje que usan los algoritmos de machine learning. Decide. <https://decidesoluciones.es/tipos-de-aprendizaje-algoritmos-machine-learning/>
- [11] Editor. (2022, 18 marzo). *Semi-Supervised Learning, explained with examples*. AltexSoft. <https://www.altexsoft.com/blog/semi-supervised-learning/>
- [12] *Types of Machine Learning*. (s.f.). JavatPoint. <https://www.javatpoint.com/types-of-machine-learning>
- [13] Gillis, A. S., Burns, E., & Brush, K. (2023). *Deep learning*. Enterprise AI. <https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network>
- [14] *¿Qué es el aprendizaje profundo? - Explicación del aprendizaje profundo - AWS*. (s.f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/deep-learning/#:~:text=Deep%20learning%20is%20a%20method,produce%20accurate%20insights%20and%20predictions.>
- [15] Castillo, E. (1998). *Introducción a las redes funcionales con aplicaciones: un nuevo paradigma neuronal*. Paraninfo.

- [16] García, J. D. V. (2020, 23 octubre). Redes neuronales desde cero (I) - Introducción - IArtificial.net. *IArtificial.net*. <https://www.iartificial.net/redes-neuronales-desde-cero-i-introduccion/>
- [17] Mishra, M. (2021, 15 diciembre). *Convolutional neural networks, explained - towards data science*. Medium. <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>
- [18] *CS 230 - Recurrent Neural Networks Cheatsheet*. (s.f.). <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>
- [19] GeeksforGeeks. (2023). *Introduction to recurrent neural network*. *GeeksforGeeks*. <https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/>
- [20] Brownlee, J. (2019). *A gentle introduction to Generative Adversarial networks (GANs)*. MachineLearningMastery.com. <https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/>
- [21] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). *Generative adversarial nets*. Advances in neural information processing systems, 27.
- [22] Dwivedi, H. (2023). *Understanding GAN loss functions*. neptune.ai. <https://neptune.ai/blog/gan-loss-functions>
- [23] Mirza, M., & Osindero, S. (2014). *Conditional generative adversarial nets*. arXiv preprint arXiv:1411.1784.
- [24] Nayak, M. (2021, 9 diciembre). *An Introduction to Conditional GANs (CGANs)*. DataDrivenInvestor. <https://medium.datadriveninvestor.com/an-introduction-to-conditional-gans-cgans-727d1f5bb011>
- [25] Arjovsky, M., Chintala, S., & Bottou, L. (2017, July). *Wasserstein generative adversarial networks*. In International conference on machine learning (pp. 214-223). PMLR.
- [26] K, B. (2022). *WGAN: Wasserstein Generative Adversarial Networks*. Paperspace Blog. <https://blog.paperspace.com/wgans/>
- [27] Google. (s.f.). Captura de pantalla de Google Earth. Recuperada de <https://www.google.com/earth/>
- [28] *Average monthly rainfall and snow in Nykøbing Mors (Nordjylland), Denmark (millimeter)*. (s.f.). World Weather & Climate Information. <https://weather-and-climate.com/average-monthly-precipitation-Rainfall,nykobing-mors-nordjylland-dk,Denmark>
- [29] Scher, S., & Peßenteiner, S. (2021). *Technical note: Temporal disaggregation of spatial rainfall fields with generative adversarial networks*. Hydrology and Earth System Sciences, 25(6), 3207-3225. <https://doi.org/10.5194/hess-25-3207-2021>
- [30] J. Hany and G. Walters, Hands-On Generative PyTorch 1 . x.

- [31] Abascal, M. Á. H. (2023). *Generative adversarial networks: a rivalry that strengthens*. Quantdare. <https://quantdare.com/generative-adversarial-networks-a-rivalry-that-strengthens/>
- [32] Sinani, A. (2022, 30 marzo). *Don't be intimidated it's only machine learning*. Medium. <https://medium.com/@artin.sinani/dont-be-intimidated-it-s-only-machine-learning-158f703ab387>
- [33] Tekmeteo. (2023). *Modelo ECWF - Centro Europeo de Predicciones Meteorológicas a Plazo Medio*. Tekmeteo. <https://tekmeteo.com/centro-europeo-de-previsiones-meteorologicas-a-plazo-medio>
- [34] *ERA5 atmospheric reanalysis — Climate Data Guide*. (1940, January 1). NCAR. <https://climatedataguide.ucar.edu/climate-data/era5-atmospheric-reanalysis>
- [35] Müller, H. and Haberlandt, U. (2018). *Temporal Rainfall Disaggregation Using a Multiplicative Cascade Model for Spatial Application in Urban Hydrology*. *J. Hydrol.*, 556, 847–864, <https://doi.org/10.1016/j.jhydrol.2016.01.031>
- [36] Kumar, J., Brooks, B.-G. J., Thornton, P. E., and Dietze, M. C. (2012) *Sub-daily Statistical Downscaling of Meteorological Variables Using Neural Networks*. *Proced. Comput. Sci.*, 9, 887–896, <https://doi.org/10.1016/j.procs.2012.04.095>
- [37] Koutsoyiannis, D. and Onof, C. (2001) *Rainfall Disaggregation Using Adjusting Procedures on a Poisson Cluster Model*. *J. Hydrol.*, 246, 109–122, [https://doi.org/10.1016/S0022-1694\(01\)00363-8](https://doi.org/10.1016/S0022-1694(01)00363-8)
- [38] Sharma, A. and Srikanthan, S. (2006) *Continuous Rainfall Simulation: A Nonparametric Alternative*, in: *30th Hydrology & Water Resources Symposium: Past, Present & Future*, 4–7 December 2006, Launceston, Tasmania, p. 86