Escuela de Doctorado
y Estudios de Posgrado
Universidad de La Laguna

Trabajo de Fin de Máster

Detección de eventos climáticos mundiales

Detection of global weather events

Omar Patricio Pérez Znakar

La Laguna, 4 de marzo de 2024

D. **Vicente José Blanco Pérez**, con N.I.F. 42171808C profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor.

D. **Francisco Carmelo Almeida Rodriguez**, con N.I.F. 42831571M Catedrático de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

"*Detección de eventos climáticos mundiales*"

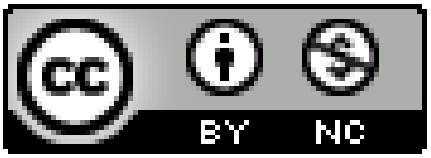
ha sido realizada bajo su dirección por D. **Omar Patricio Pérez Znakar**, con N.I.F. 79062976Q.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 4 de marzo de 2024

Agradecimientos

Quiero agradecer este periodo de investigación y aprendizaje constante, no sólo en límites de conocimiento empírico sino también en crecimiento personal. Por esta razón, en primera instancia me gustaría agradecer a mis tutores D.Vicente José Blanco Pérez y a D.Francisco Carmelo Almeida Rodriguez por toda la dedicación y entrega que me han aportado en este periodo, tanto en recursos como en tiempo. Asimismo, también me gustaría agradecer a mi familia, ya que han sido pilares sin los cuales no podría haber llegado a este momento

Licencia



Resumen

El cambio climático representa uno de los desafíos más críticos y urgentes de nuestra era, con los shocks climáticos —eventos extremos como olas de calor, huracanes, y sequías— emergiendo como manifestaciones alarmantes de este fenómeno global. La capacidad para detectar y analizar de manera precisa estos eventos no solo es fundamental para la investigación científica, sino también para la formulación de políticas de adaptación y mitigación efectivas. En este contexto, el proyecto actual busca avanzar significativamente en la detección de shocks climáticos a través del desarrollo y aplicación de tecnologías de vanguardia en inteligencia artificial y procesamiento de grandes volúmenes de datos.

Este estudio se centra en la integración de algoritmos avanzados de aprendizaje automático y técnicas de procesamiento de datos masivos para mejorar la identificación y predicción de eventos climáticos extremos. Utilizando una amplia gama de datos meteorológicos, se propone evaluar la efectividad de diferentes algoritmos de detección de anomalías en la identificación de patrones climáticos significativos. La paralelización de estos procesos en entornos de computación de altas prestaciones se investiga como un medio para acelerar el análisis y mejorar la eficiencia del procesamiento de datos.

La adopción de Kubernetes y otras tecnologías de contenerización juega un papel crucial en este proyecto, permitiendo la gestión eficiente y escalable de cargas de trabajo computacionales distribuidas. Este enfoque no solo facilita un análisis más rápido y flexible de grandes conjuntos de datos, sino que también abre nuevas posibilidades para la implementación de soluciones de análisis climático en tiempo real. Al combinar estas tecnologías avanzadas, el proyecto aspira a establecer nuevos estándares en la detección y análisis de shocks climáticos, proporcionando herramientas valiosas para científicos, responsables políticos y comunidades afectadas por el cambio climático.

La culminación de este trabajo espera no solo avanzar en el conocimiento científico sobre los fenómenos climáticos extremos sino también ofrecer aplicaciones prácticas para la mitigación de sus efectos y la adaptación a un clima cambiante. Al mejorar la capacidad para predecir y responder a shocks climáticos, este proyecto contribuye a la resiliencia global frente al cambio climático, marcando un paso importante hacia la sostenibilidad y la protección de ecosistemas y comunidades vulnerables en todo el mundo.

Palabras clave: Cambio climático, Shocks climáticos, Detección de anomalías, Procesamiento de datos masivos, Inteligencia artificial, Aprendizaje automático, Kubernetes, Paralelización de procesos y Análisis climático

Abstract

Climate change represents one of the most critical and urgent challenges of our era, with climate shocks—extreme events such as heatwaves, hurricanes, and droughts—emerging as alarming manifestations of this global phenomenon. The ability to accurately detect and analyze these events is not only fundamental for scientific research but also for the formulation of effective adaptation and mitigation policies. In this context, the current project seeks to make significant advancements in the detection of climate shocks through the development and application of cutting-edge technologies in artificial intelligence and big data processing.

This study focuses on integrating advanced machine learning algorithms and massive data processing techniques to enhance the identification and prediction of extreme climatic events. Using a wide range of meteorological data, it aims to evaluate the effectiveness of different anomaly detection algorithms in identifying significant climatic patterns. The parallelization of these processes in high-performance computing environments is investigated as a means to accelerate analysis and improve data processing efficiency.

The adoption of Kubernetes and other containerization technologies plays a crucial role in this project, enabling the efficient and scalable management of distributed computational workloads. This approach not only facilitates faster and more flexible analysis of large datasets but also opens up new possibilities for the implementation of real-time climatic analysis solutions. By combining these advanced technologies, the project aspires to set new standards in the detection and analysis of climate shocks, providing valuable tools for scientists, policymakers, and communities affected by climate change.

The culmination of this work hopes not only to advance scientific knowledge about extreme climatic phenomena but also to offer practical applications for mitigating their effects and adapting to a changing climate. By improving the ability to predict and respond to climate shocks, this project contributes to global resilience against climate change, marking an important step towards sustainability and the protection of ecosystems and vulnerable communities worldwide.

Keywords: *Climate change, Climate shocks, Anomaly detection, Big data processing, Artificial intelligence, Machine learning, Kubernetes, Process parallelization and Climate analysis*

Índice general

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

1.1. Descripción y objetivos

En el contexto contemporáneo, el cambio climático emerge como una de las amenazas más formidables a la estabilidad y resiliencia de nuestros sistemas naturales y humanos. La incidencia de shocks climáticos, caracterizados por su extrema severidad y anomalías meteorológicas, desencadena una cadena de efectos adversos con el potencial de socavar la seguridad alimentaria, deteriorar la biodiversidad y comprometer el sustento económico de comunidades a nivel global. Frente a este panorama, se vuelve imperativo el desarrollo e implementación de metodologías avanzadas para la detección y análisis profundo de estos fenómenos. La capacidad de anticipar y mitigar estos eventos extremos se posiciona como un pilar esencial en la estrategia global de adaptación y mitigación del cambio climático.

Este estudio propone un enfoque novedoso que integra técnicas de procesamiento de datos avanzadas y algoritmos de inteligencia artificial para elevar la precisión y eficacia en la detección y predicción de shocks climáticos. Mediante la exploración exhaustiva de amplios conjuntos de datos meteorológicos, este trabajo se enfoca en:

- **Elaboración de un Marco Metodológico Integral:** este objetivo contempla la creación de un sistema sofisticado que amalgama técnicas de análisis estadístico de vanguardia, aprendizaje automático y herramientas de visualización de datos. Este enfoque tiene como finalidad la identificación precisa y oportuna de eventos climáticos extremos, superando las capacidades de los sistemas convencionales de detección.
- **Evaluación Rigurosa de Algoritmos de Detección de Anomalías:** se llevará a cabo una evaluación de diversos algoritmos, poniendo especial énfasis en aquellos fundamentados en el aprendizaje profundo, con el objetivo de determinar los más eficaces para discernir patrones climáticos significativos. Este análisis comparativo pretende no solo identificar, sino también refinar las herramientas analíticas, estableciendo un nuevo estándar de precisión y velocidad en la detección de anomalías climáticas.

La realización de este ambicioso proyecto requiere de una metodología rigurosa, que incluye:

- **La Identificación Avanzada de Shocks Climáticos:** empleo de técnicas sofisticadas de análisis de series temporales y reconocimiento de patrones para la detección temprana de eventos climáticos críticos, representando un avance significativo en la monitorización ambiental.
- **Optimización del Procesamiento de Datos Climáticos a Gran Escala:** evaluación y aplicación de técnicas de procesamiento paralelo y distribuido para el manejo eficiente de vastas bases de datos climáticas. Incluye el uso de tecnologías como Kubernetes para facilitar el análisis distribuido y escalable, asegurando un aprovechamiento óptimo de los recursos computacionales.
- **Integración de Técnicas de Procesamiento Eficiente en el Flujo de Trabajo de Investigación:** establecimiento de un protocolo que integre la técnica de procesamiento paralelo más efectiva, permitiendo la detección y análisis exhaustivo de shocks climáticos en tiempos reducidos. Este enfoque busca acelerar la eficiencia de la investigación y maximizar la relevancia y aplicabilidad de los resultados.

Con este marco de trabajo, el estudio se propone no solo contribuir significativamente al campo de la ciencia climática, sino también ofrecer herramientas prácticas para la toma de decisiones en materia de política climática y gestión de riesgos ambientales. A través de la integración de avances tecnológicos y colaboraciones estratégicas, se espera que los resultados de esta investigación marquen un hito en la lucha contra el cambio climático, sentando las bases para un futuro más resiliente y sostenible.

1.2. Justificación del proyecto

La creciente incidencia y severidad de los eventos climáticos extremos, exacerbados por el cambio climático global, subraya la urgencia de desarrollar métodos avanzados para su detección y análisis. Este proyecto propone avanzar

en la investigación actual, ofreciendo nuevas técnicas y metodologías basadas en inteligencia artificial y procesamiento de datos a gran escala. La justificación de este proyecto se fundamenta en la necesidad imperante de mejorar las capacidades de predicción y respuesta a los shocks climáticos, facilitando así estrategias más efectivas de mitigación y adaptación.

La innovación en la monitorización

y análisis de datos climáticos no solo es fundamental para avanzar en la ciencia climática, sino que también tiene el potencial de influir significativamente en la planificación de políticas públicas y la gestión de riesgos naturales. Este proyecto se distingue por su enfoque interdisciplinario, combinando experiencia

en climatología, ciencia de datos, y tecnología de la información para abordar un problema complejo y multifacético. Además, la implementación de este estudio contribuirá al conocimiento científico, ofreciendo una base sólida para futuras investigaciones y aplicaciones prácticas en la lucha contra el cambio climático.

1.3. Antecedentes y Estado Actual

El estudio de los shocks climáticos y la detección de anomalías ha evolucionado significativamente en las últimas décadas, marcado por investigaciones fundamentales que han establecido un sólido entendimiento del impacto del cambio climático en eventos meteorológicos extremos. Dentro de este contexto, es esencial destacar investigaciones representativas que han contribuido a este campo.

Una de las investigaciones pioneras, que analiza las precipitaciones en Falla, Cuba, durante un periodo de 30 años, implementó métodos estadísticos para identificar cambios climáticos a nivel local "El cambio climático y sus evidencias en las precipitaciones" [8]. Este estudio es crucial, ya que demuestra la aplicabilidad de técnicas estadísticas en la detección de variaciones climáticas y establece un precedente para la evaluación de anomalías climáticas mediante el análisis de series temporales.

Adicionalmente, el informe de la Organización Meteorológica Mundial y la Oficina de las Naciones Unidas para la Reducción del Riesgo de Desastres [34] es fundamental para comprender la dinámica de las catástrofes relacionadas con el clima a lo largo de los últimos 50 años ("Las catástrofes relacionadas con el clima se quintuplican en 50 años"). Este documento no solo resalta el aumento de eventos climáticos extremos impulsados por el cambio climático sino también el impacto positivo de los sistemas de alerta temprana en la reducción de mortalidades, subrayando la importancia de la detección precoz y la gestión de riesgos.

En el ámbito de la innovación técnica, el proyecto de investigación sobre "Detección de anomalías y Análisis en tiempo real" [9]

del instituto DaSCI ilustra el desarrollo de algoritmos avanzados para la detección de anomalías. Esta línea de investigación resalta la versatilidad de estas técnicas, no solo en el contexto climático sino también en aplicaciones que van desde la salud hasta la industria, demostrando el potencial transversal de estas metodologías para abordar una amplia gama de desafíos.

Estas referencias clave ilustran tanto los avances como los desafíos persistentes en la detección y análisis de shocks climáticos. A pesar de los progresos realizados, la necesidad de mejorar las técnicas de detección de anomalías, junto con el desarrollo de sistemas de monitoreo más eficientes, sigue siendo evidente. La continua evolución de este campo sugiere un potencial significativo para futuras investigaciones que pueden construir sobre estos fundamentos para desarrollar soluciones más sofisticadas y eficaces en la lucha contra los impactos del cambio climático.

Capítulo 2

Conjunto de datos utilizados

En esta sección, exploraremos los diversos tipos de datos meteorológicos recopilados de fuentes a nivel mundial, enfocándonos específicamente en el conjunto de datos seleccionado para nuestro proyecto. Es importante señalar que estos datos provienen de registros meteorológicos globales [15, 33] garantizando así su autenticidad y relevancia para el análisis climático.

Los datos se han organizado en diversas categorías clave, incluyendo longitud, latitud, altitud, mes, año, así como diferentes medidas de temperatura (mínima, máxima y media) y precipitación. A continuación, ofrecemos detalles sobre cada una de estas categorías, comenzando por la longitud y la latitud.

- **Longitud y latitud:** estos dos parámetros geográficos son fundamentales, ya que nos permiten localizar con precisión las estaciones meteorológicas de las que se han obtenido los datos. Para ilustrar mejor esta información, presentaremos un mapa que muestra la ubicación de las estaciones meteorológicas incluidas en nuestro conjunto de datos.

- **Altitud:** representa la distancia vertical desde un punto específico hasta el nivel del mar. Este parámetro es crucial para entender cómo varían las condiciones meteorológicas con la altitud.

- **Mes:** el conjunto de datos abarca todos los meses del año, desde enero hasta diciembre, proporcionando una visión completa de las variaciones estacionales.

- **Años:**: los datos incluidos en el estudio cubren un amplio rango temporal, específicamente desde 1964 hasta 2017. Esto permite analizar tendencias climáticas a largo plazo.

- **Temperatura:** este parámetro mide el calor o frío de un cuerpo o ambiente, siendo una variable esencial en los estudios meteorológicos. Se detalla en tres categorías principales:

    - **Mínima:** refiere a la temperatura más baja registrada en un mes. Es un indicador clave para entender las condiciones más frías que pueden presentarse en una región específica. Para ilustrar este concepto, se incluirá

un gráfico que muestra la evolución de la temperatura mínima en una región determinada a lo largo del tiempo.

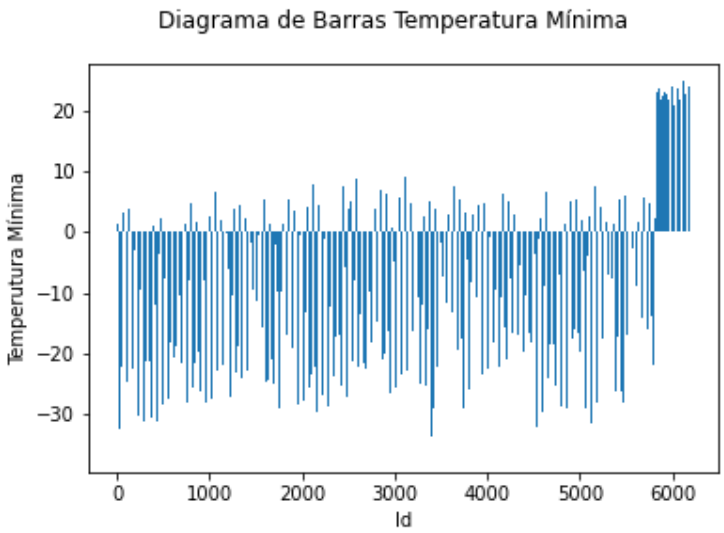


Figura 2.1: Evolución de la temperatura mínima en una región.

- **Máxima:** esta categoría representa la temperatura más alta alcanzada durante un mes. Es un indicador crucial para evaluar las condiciones de calor extremo en una región específica. La siguiente figura ilustrará cómo varía este parámetro, mostrando su evolución en una zona determinada a lo largo de un periodo establecido.

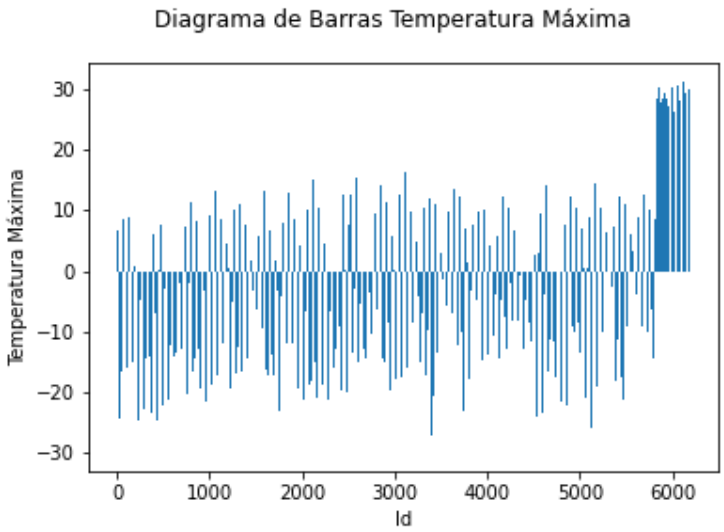


Figura 2.2: Evolución de la temperatura máxima en una región.

- **Media:** la temperatura media se calcula como el promedio de las temperaturas diarias durante un mes completo, ofreciendo una visión equilibrada de las condiciones climáticas generales de una región. Este indicador es esencial para entender el clima promedio y las variaciones estacionales en una zona específica. A continuación, presentaremos un gráfico que representa la evolución de la temperatura media en una región seleccionada, proporcionando una visualización clara de cómo fluctúa este parámetro a lo largo del tiempo.

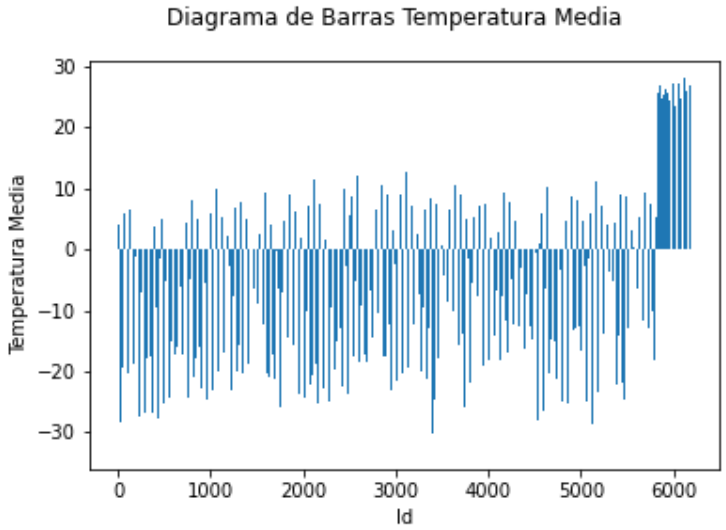


Figura 2.3: Evolución de la temperatura media en una región.

- **Precipitación:** este parámetro cuantifica el volumen de precipitación caído durante un mes específico, siendo un indicador clave para evaluar la cantidad de lluvia, nieve, o cualquier forma de hidrometeoro que ha llegado al suelo. La medición de la precipitación es fundamental para estudios hidrológicos y meteorológicos, ya que ayuda a comprender patrones de clima y a prever posibles eventos extremos. En la siguiente representación gráfica, mostraremos cómo se distribuye la precipitación en una zona determinada, ofreciendo una visualización de las variaciones mensuales de este importante parámetro.

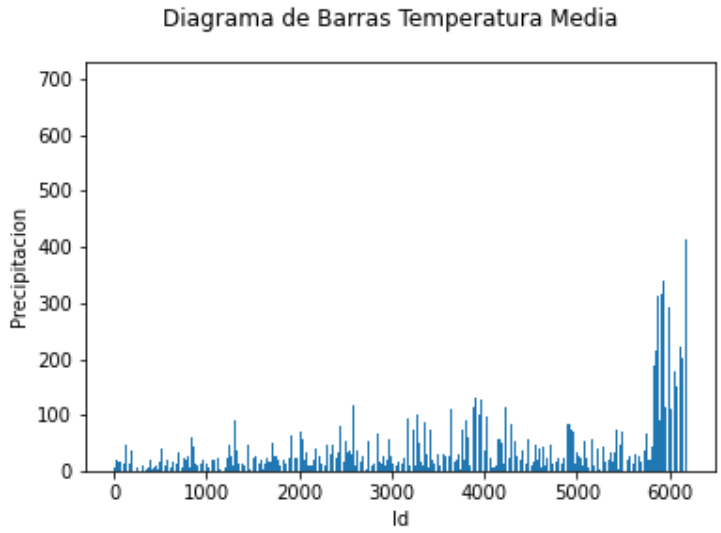


Figura 2.4: Evolución de la precipitación en una región.

Capítulo 3

Preprocesado de datos

En esta sección, nos enfocamos en el preprocesado de datos, un proceso preliminar crucial donde cada dato es meticulosamente analizado [22, 2, 32]. Este análisis incluye tareas esenciales como la limpieza de datos, su reestructuración y la identificación y eliminación de ruido, entre otros.

Es relevante destacar la implementación realizada en el lenguaje de programación Python para estas tareas, especialmente con el apoyo de dos bibliotecas fundamentales: Pandas[19] y Matplotlib[16]. Pandas se utiliza para el manejo y análisis eficiente de estructuras de datos, mientras que Matplotlib facilita la generación de gráficos a partir de los datos procesados.

3.1. Reestructuración de datos

La reestructuración de datos [6] implica organizar la información de forma más lógica y ordenada. En muchas ocasiones, encontramos que los datos no están estructurados de manera adecuada para su análisis o procesamiento posterior.

Para adaptar nuestros datos a un formato más coherente con las necesidades del estudio, se modificaron los nombres de ciertas columnas para mejorar la legibilidad. Por ejemplo, la columna "Temp-min" fue renombrada a "Temperatura-Minima". A continuación, presentamos el código utilizado para estos ajustes, así como una visualización del antes y después de este proceso de reestructuración:

```
    df2['Id'] = np.arange(1, len(df2) + 1) # Agregamos la columna Id

    # Renombramos las columnas
    df2.rename(columns={'Year':'Ano',
        "Temp_min" : "Temperatura_Minima", "Temp_max" : "Temperatura_Maxima",
        "Temp_med": "Temperatura_Media", "TEMP": "Temperatura",
        "Prep": "Precipitacion"},
    inplace=True)

    # Reordenamos las columnas
    df2 = df2[['Id','Longitud', 'Latitud', 'Altitud', 'Mes', 'Ano', '
    Temperatura_Minima', 'Temperatura_Maxima', 'Temperatura_Media', 'Temperatura
    ', 'Precipitacion']]

```

Listing 3.1: Código para la reestructuración de los datos meteorológicosl

Longitud	Latitud	Year	Mes	Temp_min	Temp_max	Altitud	Temp_med	TEMP	Prep
-179.75	71.25	1964	Enero	-34.099998	-27.5	125.490196	-30.799999	-30.6	13.6
-179.75	71.25	1964	Febrero	-34.900002	-28.1	125.490196	-31.500001	-31.3	10.0
-179.75	71.25	1964	Marzo	-32.400002	-25.1	125.490196	-28.750001	-28.1	9.9
-179.75	71.25	1964	Abril	-26.000000	-18.1	125.490196	-22.050000	-19.9	8.6
-179.75	71.25	1964	Mayo	-12.600000	-5.5	125.490196	-9.050000	-8.9	12.2

(a) Antes

Id	Longitud	Latitud	Altitud	Mes	Ano	Temperatura_Minima	Temperatura_Maxima	Temperatura_Media	Temperatura	Precipitacion
1	-179.75	71.25	125.490196	Enero	1964	-34.099998	-27.5	-30.799999	-30.6	13.6
2	-179.75	71.25	125.490196	Febrero	1964	-34.900002	-28.1	-31.500001	-31.3	10.0
3	-179.75	71.25	125.490196	Marzo	1964	-32.400002	-25.1	-28.750001	-28.1	9.9
4	-179.75	71.25	125.490196	Abril	1964	-26.000000	-18.1	-22.050000	-19.9	8.6
5	-179.75	71.25	125.490196	Mayo	1964	-12.600000	-5.5	-9.050000	-8.9	12.2

(b) Después

Figura 3.1: Antes y después de la reestructuración de los datos meteorológicos

3.2. Limpieza de datos

La limpieza de datos es un proceso esencial en el preprocesamiento que asegura la calidad y la precisión del análisis [10, 14]. Este proceso involucra la verificación minuciosa de las estructuras de datos para garantizar que cada elemento está correctamente asignado y que su valor es adecuado y coherente. Un claro ejemplo de la importancia de este paso se observa en la gestión de los datos temporales, como los meses, donde es imperativo que cada entrada sea válida y pertinente. Datos inconsistentes o erróneos, como encontrar el término ”amarillo”en una columna dedicada a los meses, deben ser identificados y eliminados para mantener la integridad del dataset.

En nuestro estudio, se realizó una detallada limpieza de datos para verificar la coherencia y la precisión de todas las entradas. Este proceso confirmó que cada campo del dataset cumplía con los criterios esperados y estaba libre de anomalías. A continuación, se presentan ejemplos visuales que evidencian la efectividad de nuestra limpieza de datos:

```
df2.Longitud.unique()
```

```
array([-179.75, -179.25, -178.75, -178.25, -177.75, -177.25, -176.75,
       -176.25, -175.75, -175.25, -174.75, -174.25, -173.75, -173.25,
       -172.75, -172.25, -171.75, -171.25, -170.75, -170.25])
```

(a) Longitud

```
df2.Latitud.unique()
```

```
array([ 71.25,  68.75,  68.25,  67.75,  67.25,  66.75,  66.25,  65.75,
        65.25, -16.75,  51.75, -43.75, -44.25,  64.75, -13.75,  63.25])
```

(b) Latitud

```
df2.Mes.unique()
```

```
array(['Enero', 'Febrero', 'Marzo', 'Abril', 'Mayo', 'Junio', 'Julio',
       'Agosto', 'Septiembre', 'Octubre', 'Noviembre', 'Diciembre'],
      dtype=object)
```

(c) Mes

```
df2.Ano.unique()
```

```
array([1964, 1965, 1966, 1967, 1968, 1969, 1970, 1971, 1972, 1973, 1974,
       1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985,
       1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996,
       1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007,
       2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017])
```

(d) Año

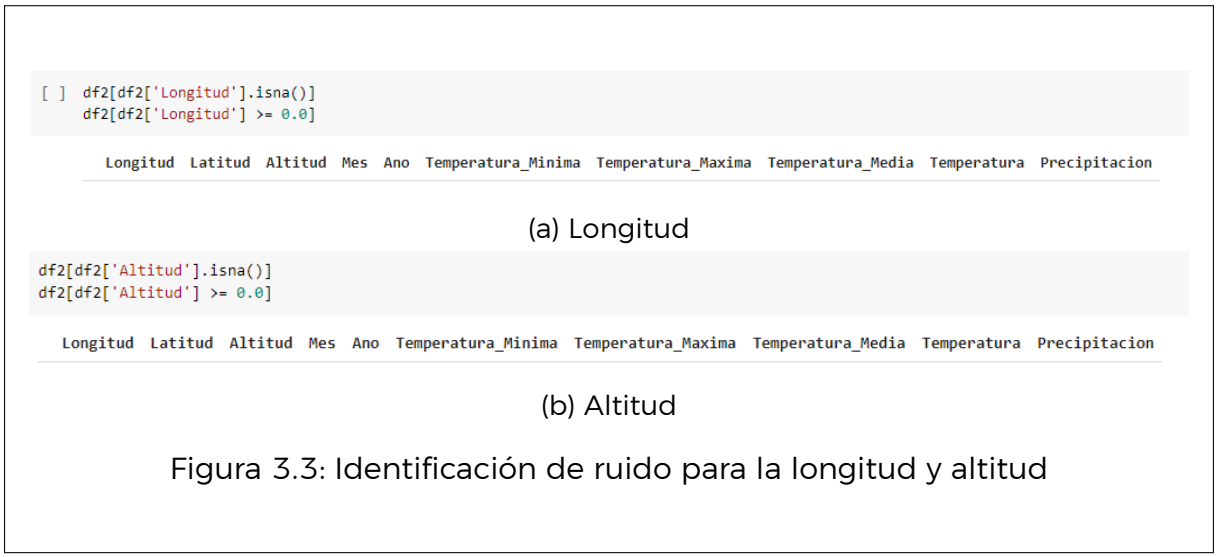
Figura 3.2: Comprobación de la limpieza de datos

3.3. Identificación de ruido

La identificación de ruido es crucial en el análisis de datos, ya que garantiza la fiabilidad de los resultados. Este proceso implica verificar que todos los datos sean válidos y lógicos, eliminando cualquier valor incongruente que pueda distorsionar el análisis. Por ejemplo, no es coherente que un valor máximo sea inferior al mínimo, o que existan duplicidades en posiciones que deberían ser únicas dentro de un conjunto de datos ordenado.

En nuestro estudio, hemos verificado exhaustivamente que todos los valores de cada columna sean racionales y pertinentes. A continuación, se detallan las comprobaciones realizadas:

- **Variables longitud y altitud:** se confirmó que todos los registros son positivos y no hay valores vacíos o nulos.



(a) Longitud

(b) Altitud

Figura 3.3: Identificación de ruido para la longitud y altitud

- **Variables latitud, mes, año, precipitación y temperatura media:** Se verificó la ausencia de valores vacíos o nulos, asegurando su coherencia.

(a) Latitud

(b) Mes

(c) Año

(d) Precipitación

(e) Temperatura Media

Figura 3.4: Identificación de ruido para diversos parámetros

- **Variable temperatura mínima:** se examinaron posibles incongruencias, como registros vacíos o nulos, y se comprobó que la temperatura mínima no superara a la máxima o a la media.

  1. La temperatura mínima sea vacía o nula.
  2. La temperatura mínima sea mayor que la máxima.
  3. La temperatura mínima sea mayor que la media.

```
df2[df2['Temperatura_Minima'].isna()]
```

Longitud	Latitud	Altitud	Mes	Ano	Temperatura_Minima	Temperatura_Maxima	Temperatura_Media	Temperatura	Precipitacion

(a) La temperatura mínima sea vacía o nula.

```
df2[df2['Temperatura_Minima'] > df2['Temperatura_Maxima']]
```

Longitud	Latitud	Altitud	Mes	Ano	Temperatura_Minima	Temperatura_Maxima	Temperatura_Media	Temperatura	Precipitacion

(b) La temperatura mínima sea mayor que la máxima.

```
df2[(df2['Temperatura_Minima'] > df2['Temperatura_Media'])]
```

Longitud	Latitud	Altitud	Mes	Ano	Temperatura_Minima	Temperatura_Maxima	Temperatura_Media	Temperatura	Precipitacion

(c) La temperatura mínima sea mayor que la media.

Figura 3.5: Identificación de ruido para la temperatura mínima

- **Variable temperatura máxima:** se revisaron los datos para descartar valores vacíos o nulos y asegurar que la temperatura máxima no fuera inferior a la temperatura media.

  1. La temperatura máxima sea vacía o nula.
  2. La temperatura máxima sea menor que la media.

```
df2[df2['Temperatura_Maxima'].isna()]
```

Longitud	Latitud	Altitud	Mes	Ano	Temperatura_Minima	Temperatura_Maxima	Temperatura_Media	Temperatura	Precipitacion

(a) La temperatura máxima sea vacía o nula.

```
df2[df2['Temperatura_Maxima'] < df2['Temperatura_Media']]
```

Longitud	Latitud	Altitud	Mes	Ano	Temperatura_Minima	Temperatura_Maxima	Temperatura_Media	Temperatura	Precipitacion

(b) La temperatura máxima sea menor que la media.

Figura 3.6: Identificación de ruido para la temperatura máxima

Cada una de estas verificaciones ha sido crucial para mantener la integridad de nuestro conjunto de datos, eliminando cualquier anomalía que pudiera afectar negativamente la calidad del análisis. A continuación, se presentan ejemplos visuales que ilustran el antes y después de la identificación y corrección de ruido en nuestro estudio:

Capítulo 4

Detección de anomalías

Antes de profundizar en el proceso de detección de anomalías[11, 27], es esencial comprender qué constituye una anomalía. Una anomalía se define como cualquier cambio o desviación de lo que se considera esperado o normal, es decir, un valor inesperado o inusual. El proceso de detección de anomalías se centra en identificar estos valores atípicos o casos raros dentro de los datos, lo cual es posible gracias a la información acumulada sobre el comportamiento habitual de estos. Con esta premisa, es importante reconocer que existen varios enfoques para la detección de anomalías, los cuales se explican a continuación:

- **Detección basada en proximidad[27]:** este método identifica como anomalías aquellos valores que presentan una distancia significativamente mayor respecto a los demás. Un valor se considera atípico si se encuentra a una distancia considerable del k-ésimo valor más cercano. Este enfoque es apreciado por su simplicidad y la facilidad de seleccionar el parámetro k. Sin embargo, su desventaja radica en la complejidad computacional elevada, específicamente $O(m^2)$, siendo m el número de objetos en el dataset.
- **Detección basada en densidad [27]:** se enfoca en identificar anomalías según la densidad de su entorno, considerando atípico aquel valor situado en áreas de baja densidad. Esta metodología presupone que la rareza de un dato es inversamente proporcional a la densidad de su área circundante. Entre las técnicas específicas dentro de este enfoque, la "inversa de la distancia" destaca por su capacidad de definir la densidad a través de un parámetro de radio preestablecido, determinando así la densidad alrededor de un punto basada en la cantidad de objetos dentro de este radio. A continuación, se profundiza en esta técnica:
    - **Inversa de la distancia:** este método calcula la densidad alrededor de un punto específico basándose en el número de objetos que caen dentro de un radio predefinido. La premisa es que un punto se considera anómalo si, dentro de este radio, se encuentra un número insuficientemente bajo de otros puntos. Esta aproximación permite una identificación efectiva de anomalías en contextos donde la proximidad y la agrupación de datos son indicativos de normalidad, contrastando fuertemente con aquellos puntos que se hallan aislados.

$$\text{densidad}(x,k) = \frac{|\mathcal{N}(x,k)|}{\sum_{y \in \mathcal{N}(x,k)} \text{dist}(x,y)}$$

donde $\mathcal{N}(x,k)$ es el conjunto de los k objetos más cercanos a x.

Figura 4.1: Formula inversa de la distancia

- **Densidad relativa [27]:** ofrece una perspectiva sofisticada para identificar anomalías, centrándose en la comparación entre la densidad de un punto específico y la densidad promedio de sus vecinos más cercanos. Esta técnica se basa en el principio de que un valor será considerado como anómalo si su densidad relativa es significativamente diferente de la de sus vecinos. En otras palabras, si un punto está considerablemente menos rodeado por otros puntos en comparación con lo que es típico en su entorno inmediato, se considera una anomalía.

$$\text{densidad\_relativa}(x,k) = \frac{\text{densidad}(x,k)}{\sum_{y \in \mathcal{N}(x,k)} \text{densidad}(y,k)/|\mathcal{N}(x,k)|}$$

Figura 4.2: Formula densidad relativa

- **Técnicas basadas en modelos[27]:** este enfoque asume que los datos atípicos son aquellos que no se ajustan bien al modelo establecido para el comportamiento normal de los datos. Utilizando técnicas basadas en la distribución de probabilidad, estos métodos evalúan la probabilidad de que cada punto pertenezca a la distribución de datos normales. Los puntos con baja probabilidad de ajuste son considerados anomalías. Esta distinción clara entre datos normales y atípicos permite categorizar los datos efectivamente, facilitando la identificación de los que se desvían significativamente de lo esperado.

El uso de técnicas basadas en modelos es particularmente útil en escenarios donde se puede definir claramente un modelo de comportamiento normal, permitiendo así detectar desviaciones significativas de este modelo. Estas técnicas son potentes para identificar anomalías en conjuntos de datos complejos y multidimensionales, donde la relación entre los datos puede ser modelada de manera efectiva.

4.1. Detección visual de los shocks climáticos

Para la realización de este apartado, es necesario aclarar que se ha utilizado la librería Plotly [21] mediante el lenguaje Python, que nos permite la visualización de una amplia gama de gráficos estadísticos. Asimismo, ofrece la posibilidad de realizar pequeños filtros en tiempo real sobre estos gráficos.

Esta sección se centra en la detección de shocks climáticos, entendidos como cambios abruptos de temperatura. Para ello, hemos empleado gráficos de líneas con dos enfoques principales: por un lado, la evolución mensual a lo largo de los años y, por otro, la evolución anual, ambos filtrados por una única zona geográfica. A continuación, ofrecemos una breve explicación de cada uno de estos gráficos:

- **Evolución mensual a lo largo de los años**: este gráfico traza la trayectoria mensual de la temperatura a lo largo de varios años. Permite la selección de variables específicas para ajustarse mejor a nuestros objetivos de análisis, tales como la temperatura (media, máxima, mínima), longitud y latitud de la cuadrícula, o incluso la visualización individual de cada mes. La figura 4.3 muestra un ejemplo de esta visualización.

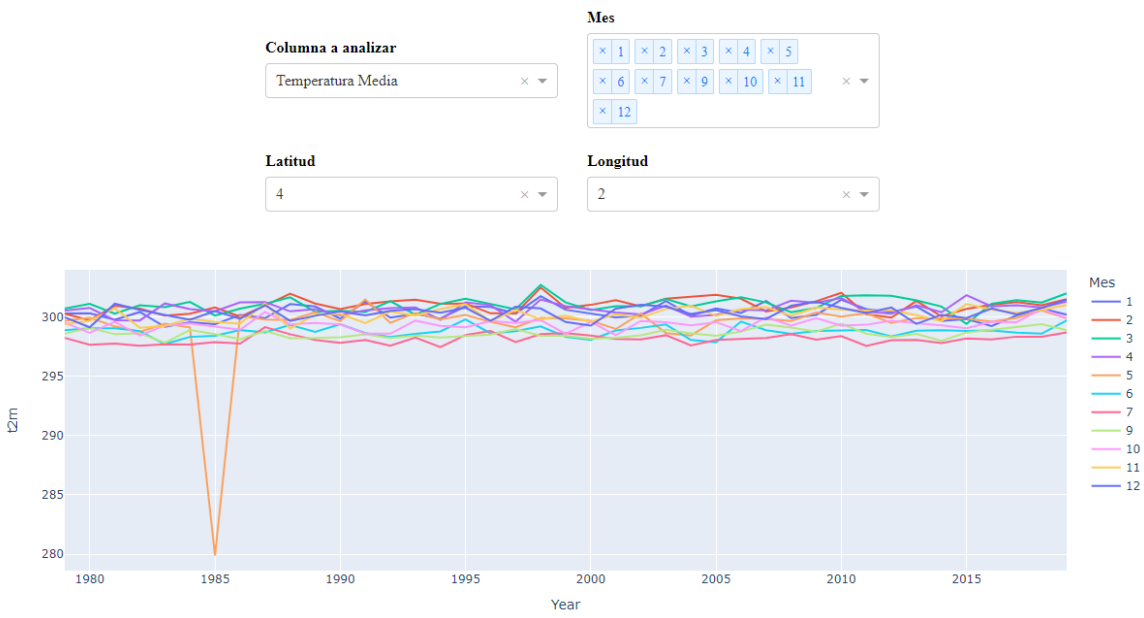


Figura 4.3: Detección Visual utilizando la evolución mensual

- **Evolución anual**: este gráfico permite comparar la variabilidad de la temperatura mes a mes, año tras año. Facilita la adaptación de los parámetros de análisis a los objetivos específicos, permitiendo la elección entre diferentes medidas de temperatura o la especificación de coordenadas geográficas. La figura 4.4 proporciona una representación de este análisis.

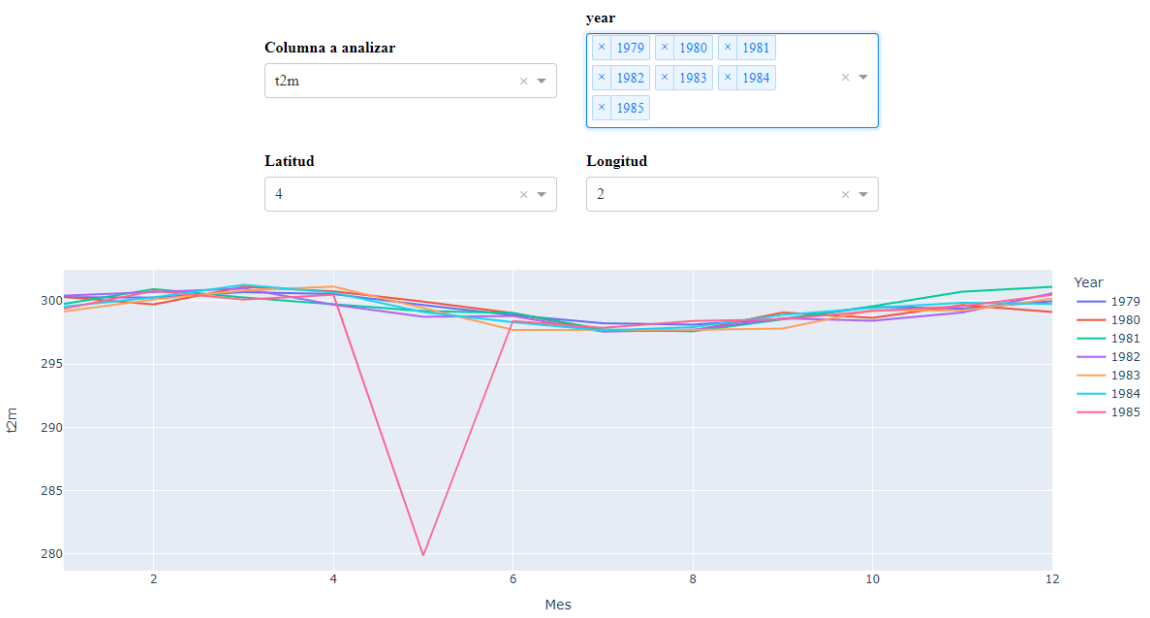


Figura 4.4: Detección Visual utilizando la evolución mensual

A través de las visualizaciones generadas por Plotly, es posible observar cómo se detecta una anomalía en el mes de mayo de 1985, evidenciada tanto en la evolución mensual como en la anual. Este descubrimiento destaca la capacidad de los algoritmos implementados para identificar anomalías, sirviendo como un proceso de verificación para evaluar la precisión y la fiabilidad de los resultados obtenidos por dichos algoritmos.

4.2. Algoritmos utilizados

A lo largo de este punto se desarrollará una breve explicación de los diferentes algoritmos desarrollados en R para la detección de anomalías sin tener en cuenta las series temporales.

4.2.1. LOF (Local Outlier Factor)

El LOF, según lo definido por Breunig, M. M., Kriegel, H.-P., Ng, R. T., y Sander, J. (2000) [4, 23, 26], es un algoritmo basado en la densidad que identifica outliers en función de la distancia a sus vecinos más cercanos. Esto significa que los valores anómalos son aquellos con una densidad significativamente menor en comparación con sus vecinos.

Este algoritmo busca calcular la anormalidad o el grado de atipicidad de los datos, señalando aquellos valores que difieren notablemente del conjunto, siendo así reconocidos como valores atípicos o ”outliers”. Es importante resaltar que hemos aplicado LOF a nuestro conjunto de datos meteorológicos, identificando aquellas condiciones climáticas excepcionales. En la figura 4.5, se presenta una

visualización de estos datos, donde el eje X muestra los outliers detectados y el eje Y la frecuencia de estos valores, destacando que cuanto más a la derecha esté un punto en el gráfico, más inusual es la condición climática que representa.

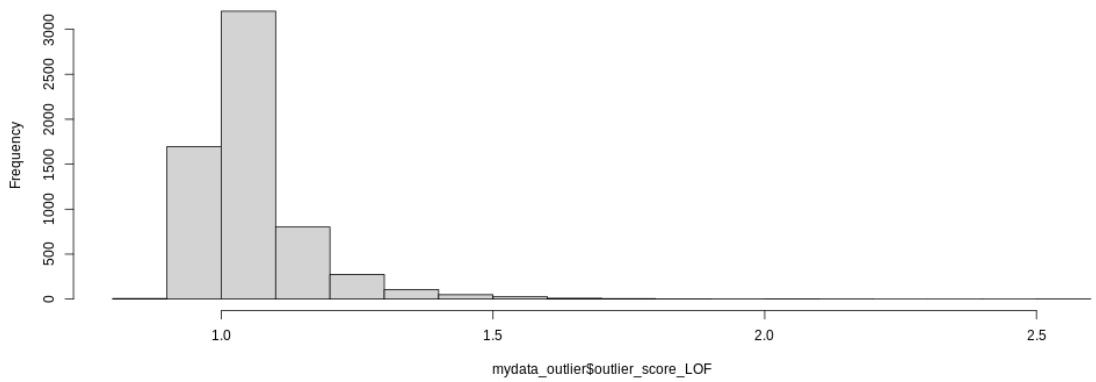


Figura 4.5: Histograma del algoritmo LOF

Adicionalmente, se ha representado gráficamente (figura 4.6) la distribución de los ”outliers” identificados mediante este algoritmo, utilizando como umbral el valor cero. Por consiguiente, aquellos datos por encima de este umbral son considerados superiores a la media, y viceversa para los valores inferiores a cero.

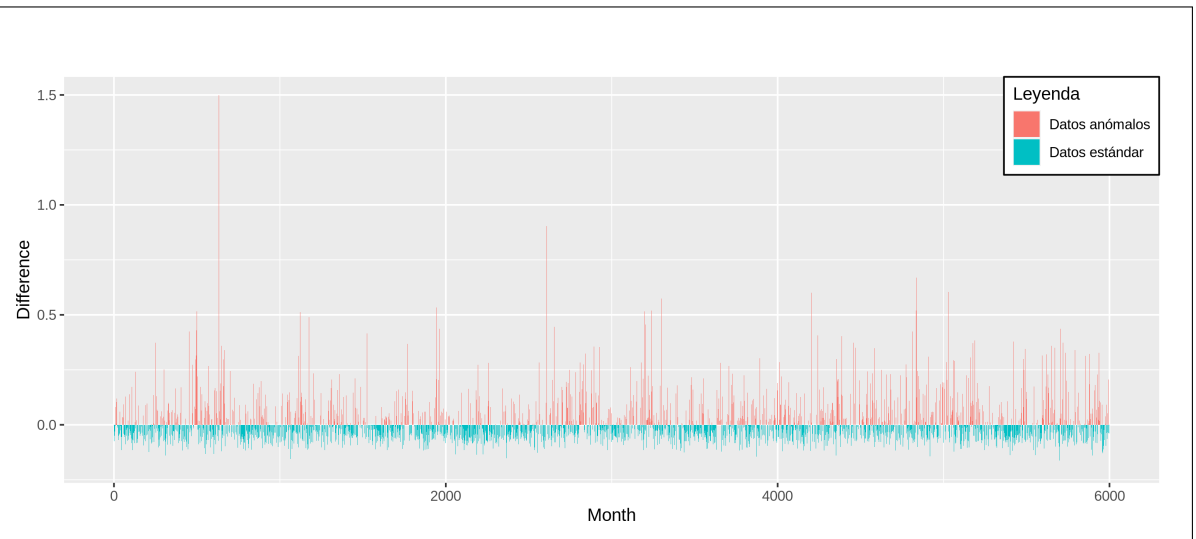


Figura 4.6: Diferencia de outliers de LOF con respecto a la media

4.2.2. LDOF (Local Distance-based Outlier Factor)

El algoritmo LDOF, según Zhang, K., Hutter, M., y Jin, H. (2009) [35, 23, 28], se enfoca en la detección de anomalías mediante el análisis de la proximidad entre los valores. Esto significa que un valor se considera atípico si se encuentra significativamente alejado de sus vecinos más cercanos.

El LDOF evalúa qué tan apartado está un objeto en relación con su entorno inmediato, determinando su grado de anomalía en función de la distancia a sus vecinos más próximos. Es relevante mencionar que, aunque la teoría detrás de este algoritmo es robusta, incluyendo análisis sobre su eficacia, límites inferiores y la probabilidad de detección errónea, su aplicación práctica es igualmente significativa. Utilizando la técnica top-n, el LDOF puede identificar de manera eficaz anomalías extremas, sobresaliendo sobre métodos convencionales como el top-n KNN y top-n LOF por su rendimiento superior y la facilidad de ajuste de parámetros. Esta técnica no solo ha demostrado ser efectiva en escenarios teóricos, sino también al aplicarla a datos reales.

El propósito de LDOF es señalar valores como anómalos basándose en su distancia respecto al conjunto general, resultando en puntuaciones que los distinguen como "outliers". Este algoritmo se aplicó a un conjunto de datos meteorológicos específico, identificando las condiciones climáticas más inusuales, como se muestra en la figura 4.7, donde el eje X indica los outliers detectados y el eje Y la frecuencia de estos eventos.

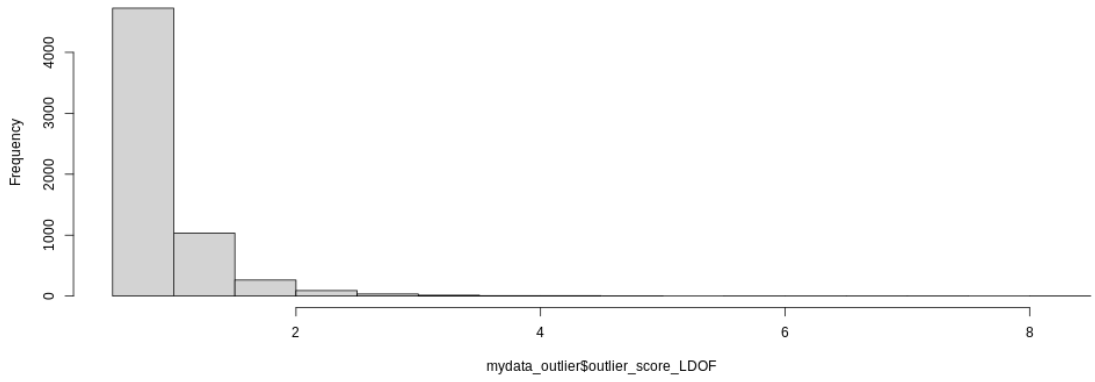


Figura 4.7: Histograma del algoritmo LDOF

La figura 4.8 ilustra la distribución de los datos anómalos identificados mediante LDOF. A diferencia de la figura anterior, esta establece el valor cero como umbral, indicando que los datos por encima de este valor se consideran superiores a la media, y viceversa para los valores inferiores.

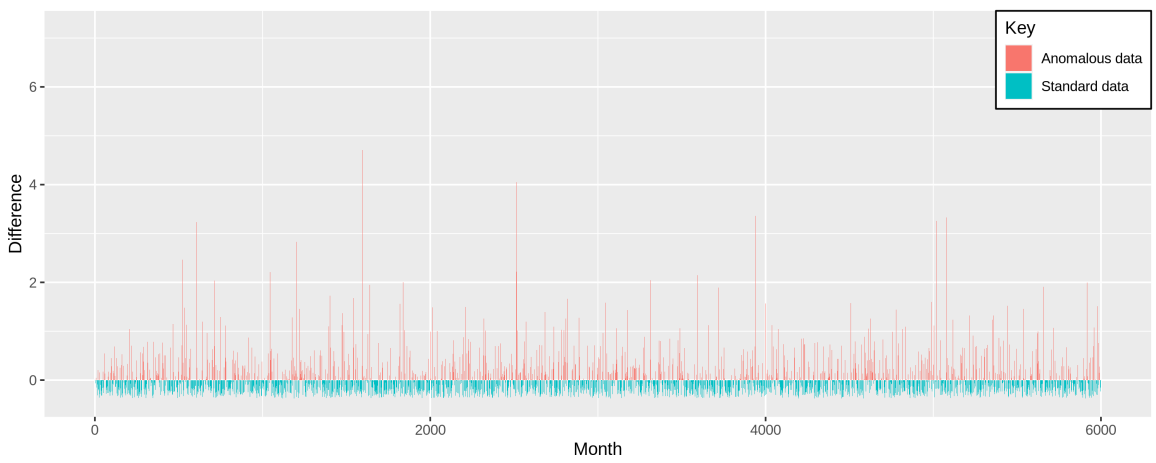


Figura 4.8: Diferencia de outliers de LDOF con respecto a la media

4.2.3. RDOS (Relative Density-based Outlier Factor)

El algoritmo RDOS, propuesto por Tang, B. y Haibo, He (2017) [23, 5], introduce un método avanzado para la detección de anomalías basado en la densidad Kernel local. Este enfoque se distingue por su capacidad para combinar la proximidad entre los puntos con el cálculo de los vecinos inversos, ofreciendo así una estimación precisa de la densidad relativa. Esta técnica permite identificar aquellos valores anómalos que se desvían significativamente de la densidad común de su entorno, marcándolos como outliers.

En el núcleo de RDOS yace el principio de que los valores atípicos son aquellos cuya densidad relativa se aparta de manera notable del conjunto general de datos. Este algoritmo es particularmente eficaz para resaltar valores dispersos, ofreciendo una herramienta robusta para reconocer y clasificar datos anómalos.

La implementación de RDOS en nuestro proyecto meteorológico se ilustra en la figura 4.9, donde se visualizan las condiciones climáticas más extremas detectadas. En este histograma, el eje X muestra los outliers identificados, mientras que el eje Y indica la frecuencia de estos eventos, destacando así aquellos valores más alejados del patrón común.

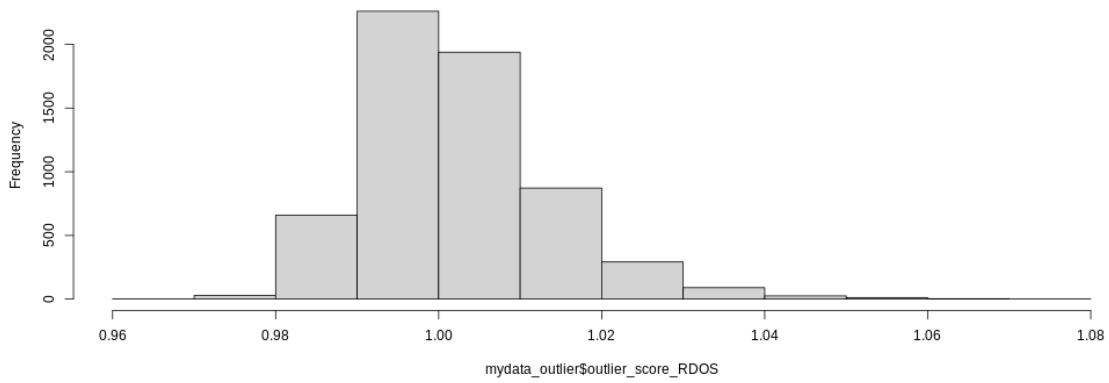


Figura 4.9: Histograma del algoritmo RDOS

Además, la figura 4.10 ofrece una visión detallada de la distribución de los outliers en relación con la media, utilizando el valor cero como referencia. Esta visualización permite distinguir de forma clara y directa aquellos datos que superan o no alcanzan el promedio general, facilitando la identificación de patrones atípicos dentro del conjunto de datos meteorológicos analizados.

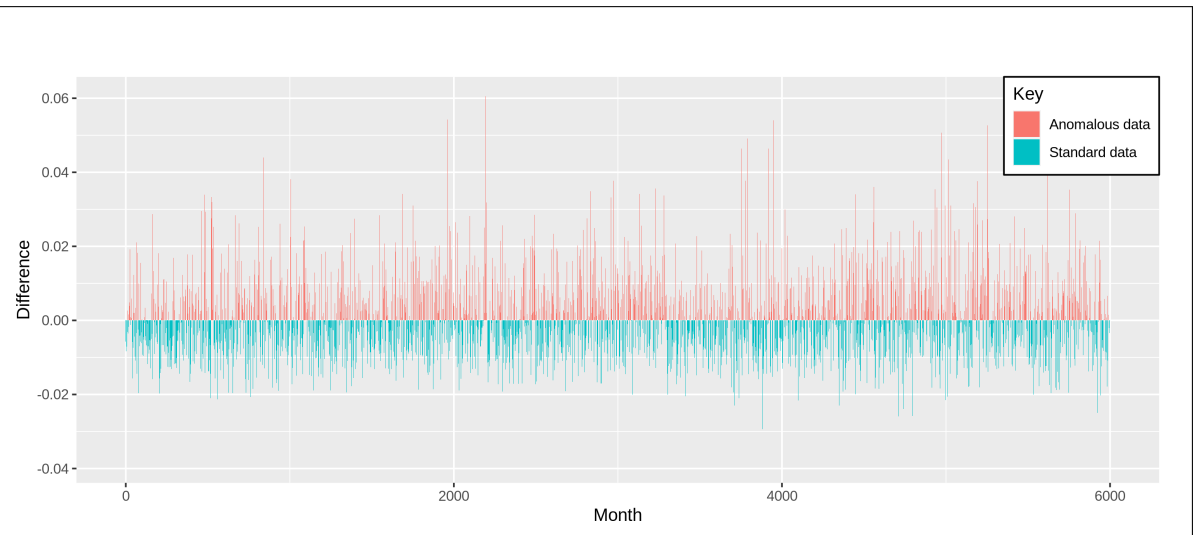


Figura 4.10: Diferencia de outliers de RDOS con respecto a la media

4.2.4. LOCI (Local Correlation Integral)

El algoritmo LOCI, propuesto por Papadimitriou, S., Gibbons, P. B., y Faloutsos, C. (2003) [23, 20], introduce un enfoque novedoso para la detección de anomalías. Este método se basa en la evaluación del número de vecinos (k) más cercanos dentro de un radio constante, identificando como anomalías aquellos datos que

se desvían de este patrón de vecindad. Una de las principales fortalezas de LOCI es su capacidad para detectar de manera eficiente no solo valores anómalos individuales sino también agrupaciones de estos, conocidas como "micro-clústeres".

Las ventajas de implementar LOCI en el análisis de datos son múltiples, destacándose las siguientes:

1. Proporciona un corte automático establecido donde se representa el punto del valor atípico, en contraste con otros métodos, donde sin ningún conocimiento se tiene que elegir los cortes.

2. Posibilita hacer gráficas para cada punto en los de datos, por lo que, aporta diferente información sobre nuestro grupo de datos: cercanía, grupos, microgrupos, distancia entre los grupos y diámetros respectivamente. Esta característica sólo la posee este algoritmo.

3. Es bastante rápido en comparación con diferentes métodos.

4. Esta metodología nos adentra hacia un método aproximado que resulta prácticamente lineal (aLOCI), lo que aporta un gran beneficio de rapidez a la hora de identificar valores anómalos con una precisión elevada. Asimismo, se debe destacar que los experimentos con esta tecnología (LOCI y aLOCI) revelan que son capaces de identificar de manera automática valores anómalos y micro-clústeres, tanto datos esperados como improvistos y sin límites para el usuario.

La figura 4.11 ilustra la aplicación de LOCI a nuestro conjunto de datos, destacando las condiciones atmosféricas más extremas identificadas. En este histograma, el eje X muestra los outliers detectados, mientras que el eje Y refleja la frecuencia de estos eventos, proporcionando una visión clara de los valores más alejados del patrón común.

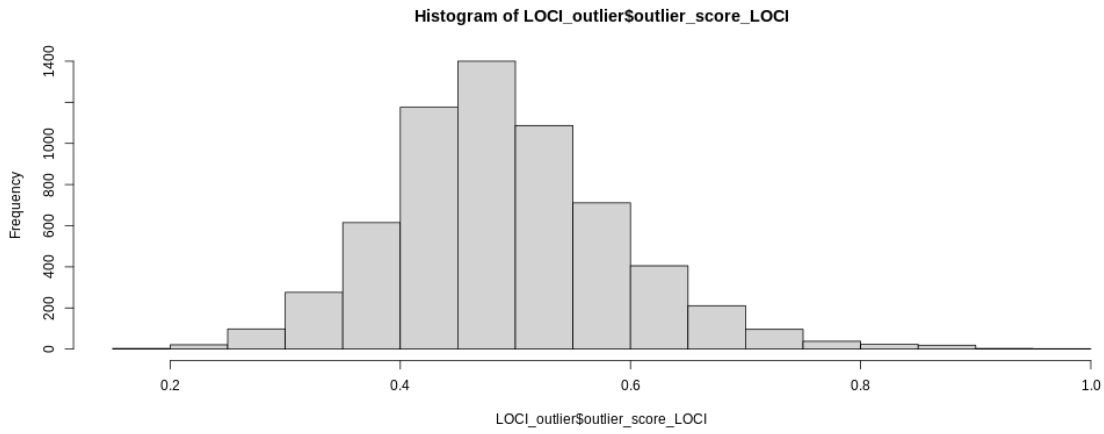


Figura 4.11: Histograma del algoritmo LOCI

Adicionalmente, la figura 4.12 muestra la distribución de los outliers en relación con la media, utilizando el valor cero como referencia. Esta visualización facilita la identificación de datos que superan o no alcanzan el promedio general, resaltando patrones atípicos dentro del análisis de datos meteorológicos.

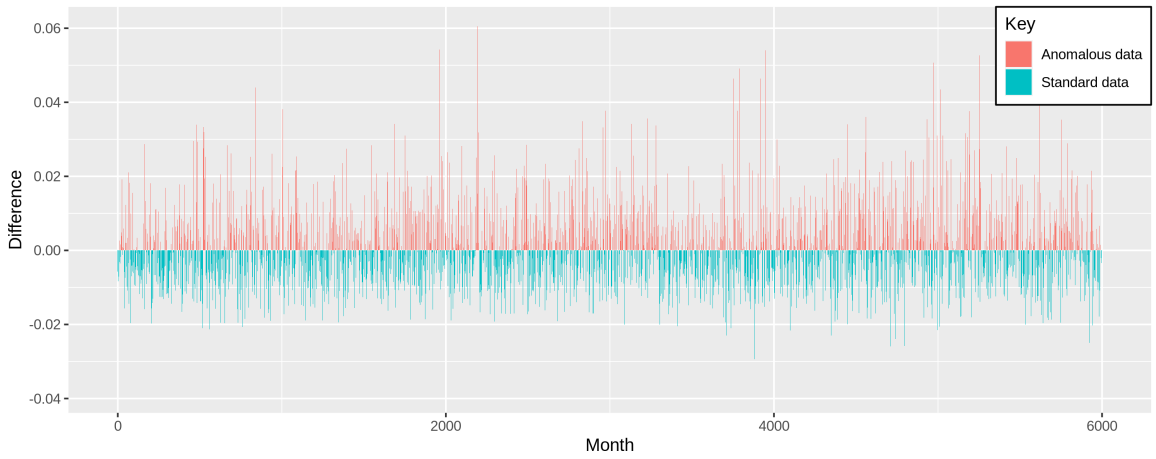


Figura 4.12: Diferencia de outliers de LOCI con respecto a la media

4.2.5. LOOP (Local Outlier Probability)

El algoritmo LOOP, definido por Kriegel, H.-P., Kröger, P., Schubert, E., y Zimek, A. (2009) [12, 23], introduce una función que estima la probabilidad de que un valor sea un outlier local. Este método basa su análisis en la densidad local de los datos, asignando a cada punto un valor entre 0 y 1, que representa la probabilidad de ser un valor atípico.

A diferencia de otros métodos para la detección de anomalías, LOOP se distingue por ofrecer una medida probabilística de la atipicidad de los datos, lo que facilita la interpretación y la toma de decisiones basada en la probabilidad de anomalía de cada punto. Este enfoque permite superar la ambigüedad inherente a otros métodos que solo proporcionan una medida de cuán anómalo es un punto sin establecer claramente su pertenencia al grupo de datos atípicos.

La figura 4.13 muestra la distribución de los outliers identificados mediante LOOP, destacando aquellos puntos que se desvían significativamente del conjunto de datos. El eje X indica los outliers detectados, mientras que el eje Y refleja la frecuencia de estos valores, evidenciando aquellos datos con mayor grado de atipicidad.

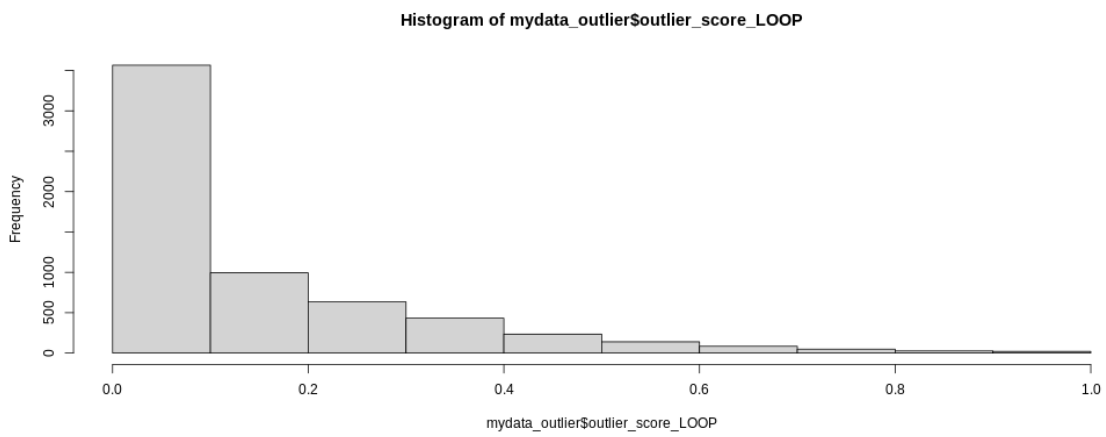


Figura 4.13: Histograma del algoritmo LOOP

Además, la figura 4.14 ilustra la distribución de los outliers en relación a la media, con el valor cero como referencia. Esta representación visual facilita la identificación de datos que exceden o no alcanzan el promedio general, permitiendo una identificación más intuitiva de patrones atípicos.

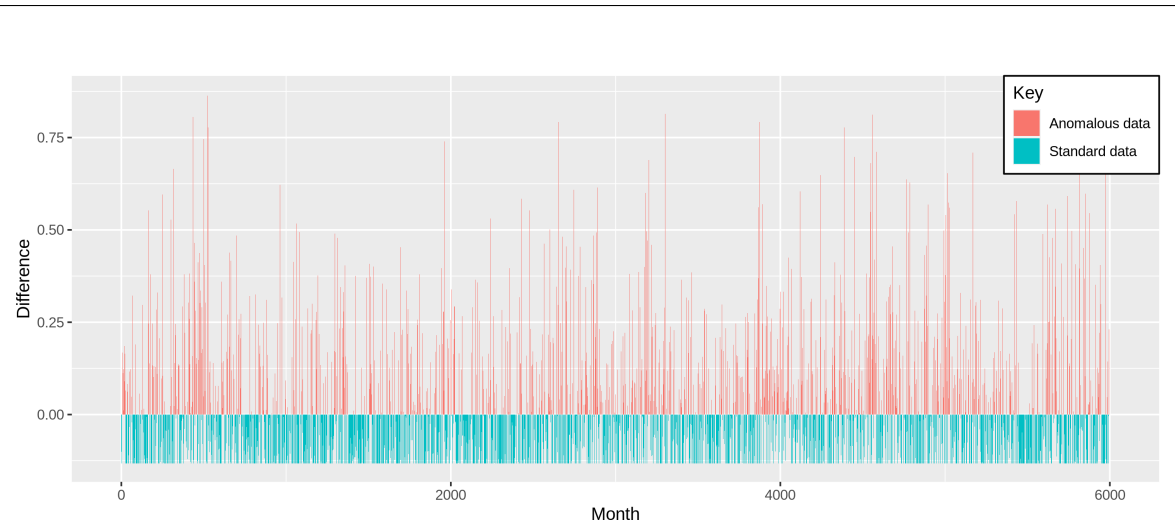


Figura 4.14: Diferencia de outliers de LOOP con respecto a la media

4.3. Cribado de datos para la aplicación del algoritmo

La selección de la estructura de datos adecuada es crucial para optimizar la detección de shocks climáticos. Se exploraron tres enfoques de cribado: temporal, mensual y atemporal, cada uno con sus propias ventajas para el análisis de datos meteorológicos. La investigación concluye que el cribado mensual ofrece el me-

jor equilibrio, eliminando información cronológica redundante mientras preserva datos esenciales para identificar anomalías según la estación del año.

- **Cribado temporal:** facilita el ordenamiento de datos en secuencia cronológica, proporcionando un marco temporal claro para el análisis.
- **Cribado mensual:** optimiza la selección de datos al preservar el mes, un elemento crítico en el análisis meteorológico, ya que la relevancia de un dato puede variar según el mes en el que se registre. Este método clasifica los datos anuales dentro de cada mes específico, facilitando la identificación de patrones o anomalías estacionales.
- **Cribado atemporal:** presenta los datos sin ningún orden cronológico, ofreciendo una perspectiva desagregada del tiempo, lo cual puede ser útil para análisis que no dependen de la secuencia temporal.

Tras un análisis meticuloso, se determina que el cribado por mes es preferible para detectar de manera efectiva los shocks climáticos. Este enfoque elimina eficientemente los datos cronológicos no esenciales, manteniendo al mismo tiempo los detalles mensuales vitales para discernir anomalías estacionales.

4.4. Selección metodología de uso de los algoritmos

Este segmento explora distintas metodologías para determinar cuál es la más adecuada para nuestro caso de estudio. Cada metodología tiene potencial para la detección de anomalías, pero es crucial seleccionar la más conveniente según nuestro objetivo específico. Las metodologías consideradas incluyen::

- **Intersección de elementos**: integra información de varios algoritmos para identificar meses coincidentes, lo que sugiere anomalías significativas.
- **Selección de un único algoritmo**: se basa exclusivamente en los resultados de un solo algoritmo para el análisis.
- **Unión de elementos**: compila datos de todos los algoritmos para incluir cualquier mes identificado como atípico, ofreciendo una visión amplia de posibles anomalías.
- **Diferencia de elementos**: utiliza todos los algoritmos relevantes, pero se enfoca en los meses que no coinciden entre ellos, proporcionando una perspectiva única sobre las discrepancias en la detección.

Dado que el objetivo principal es identificar los shocks climáticos más pronunciados, la metodología seleccionada busca armonizar los resultados de múltiples algoritmos. Este enfoque asume que los meses identificados consistentemente como atípicos por diferentes algoritmos son de particular interés, lo que permite una detección precisa y confiable de eventos climáticos extremos. La integración de varias técnicas de detección de anomalías mejora significativamente la capacidad de capturar los cambios climáticos más impactantes, asegurando una comprensión profunda y detallada de las dinámicas atmosféricas.

Capítulo 5

Técnicas de paralelización para algoritmos implementados en R

El manejo de conjuntos de datos extensos conlleva desafíos significativos en términos de eficiencia computacional y tiempo de procesamiento. En contextos donde la prontitud de los resultados es crucial, la paralelización emerge como una estrategia esencial para optimizar el uso de los recursos computacionales. Esta técnica implica dividir el código en múltiples tareas que pueden ejecutarse simultáneamente, aprovechando así todos los núcleos disponibles en el procesador para acelerar la ejecución. Nuestro enfoque se centra en aplicar la paralelización al análisis de detección de anomalías, con el objetivo de identificar el algoritmo más efectivo dentro del entorno de programación R para esta tarea.

5.1. Métodos de paralelización en R

El propósito de este segmento es explorar los distintos métodos de paralelización disponibles en R, con el objetivo de optimizar el procesamiento de grandes conjuntos de datos y, consecuentemente, reducir los tiempos de ejecución. La paralelización del código permite maximizar el uso de los recursos computacionales, especialmente los múltiples núcleos del procesador, lo cual es esencial para agilizar operaciones complejas como la detección de anomalías.

Actualmente, R ofrece una variedad de herramientas para implementar la paralelización, entre las que destacamos algunas de las más efectivas y comúnmente utilizadas en la práctica. A continuación, se presenta un resumen y una breve descripción de cada una de estas herramientas.

- **DoParrallel [30, 17]** : este conjunto de herramientas se basa en la iteración foreach para dividir y optimizar el procesamiento de tareas. DoParallel ofrece dos modalidades: una secuencial ('do') y otra paralela ('doParallel'), esta última distribuye el trabajo entre los diferentes núcleos de la CPU para mejorar la eficiencia del procesamiento.

- **Mcapply [25, 31]** exclusiva para sistemas UNIX, mclapply facilita la creación de subprocesos individuales para partes específicas de una tarea global. Al concluir estos subprocesos, el programa principal recopila y sintetiza los resultados. La cantidad de subprocesos generados corresponde al número de

núcleos disponibles en la CPU, permitiendo una paralelización efectiva del trabajo.

- **ParSapply [24]** : esta herramienta se enfoca en paralelizar la función 'sapply' en R. Requiere la configuración de un clúster virtual con especificaciones técnicas precisas, incluyendo el número de núcleos, los paquetes necesarios para la ejecución del código y, en algunos casos, credenciales para bases de datos.

- **ParLapply [24, 29, 7]** : similar a parSapply, parLapply paraleliza la función 'lapply'. Opera mediante un clúster que organiza y distribuye la carga de trabajo, basándose en una serie de parámetros como el número de núcleos, los servicios y paquetes requeridos para el análisis.

Estas herramientas representan soluciones viables para enfrentar el desafío de procesar eficientemente datos de gran volumen en R. Sin embargo, la selección de la técnica más adecuada depende de la naturaleza específica del problema a resolver. Por ello, se analizarán en detalle estas tecnologías para determinar cuál ofrece la mejor adaptabilidad y rendimiento para nuestra necesidad particular de detección de anomalías, priorizando siempre la obtención de resultados en el menor tiempo posible.

5.2. Infraestructura de ejecución

La investigación enfocada en la identificación y detección de shocks climáticos requiere de una infraestructura de ejecución robusta y eficiente. Para este fin, se dispone de una infraestructura técnica avanzada y se plantea una estructura lógica específica para optimizar el proceso de análisis.

5.2.1. Especificaciones Técnicas del Servidor.

La infraestructura se apoya en el servidor "Verode21", perteneciente al equipo de investigación de "Computación de altas prestaciones". Este servidor está equipado con dos procesadores Intel® Xeon® CPU Gold 6230N, cada uno con 20 núcleos, sumando un total de 40 núcleos que utilizan memoria compartida. Estas especificaciones técnicas subrayan la capacidad del servidor para manejar cargas de trabajo intensivas y procesamiento paralelo.

5.2.2. Implementación con Kubernetes

La estrategia central para la ejecución se basa en la implementación de Kubernetes[13, 18, 3], una tecnología líder en la gestión de contenedores. Kubernetes facilita la creación de contenedores con especificaciones técnicas ajustadas a las necesidades del análisis, permitiendo una gestión eficiente y flexible de los recursos. Los contenedores actúan como unidades de trabajo independientes, similar a máquinas virtuales, pero con la ventaja de ser más ligeros y eficientes, ya que solo instalan las dependencias necesarias para la tarea específica. Esto resulta en ahorros significativos de tiempo y recursos.

- **Eficiencia y Ligereza:** Kubernetes permite el uso de versiones muy ligeras de Linux, instalando únicamente lo necesario para cada tarea, optimizando así los tiempos de ejecución.
- **Facilidad de Despliegue:** la creación y gestión de contenedores se simplifican, permitiendo un despliegue rápido y eficaz con solo ejecutar un comando.
- **Sistema Distribuido:** la tecnología facilita la creación de sistemas distribuidos, permitiendo una escalabilidad y flexibilidad superior en el procesamiento de datos.

La figura 5.1 muestra la estructura de Kubernetes diseñada para este proyecto. Se prevé la creación de contenedores individuales para cada prueba de rendimiento, utilizando diferentes cantidades de núcleos (1, 2, 4, 8, 16 y 32). Cada contenedor estará configurado con las limitaciones de hardware correspondientes y contendrá los scripts, paquetes y librerías necesarios para ejecutar el análisis de detección de anomalías.

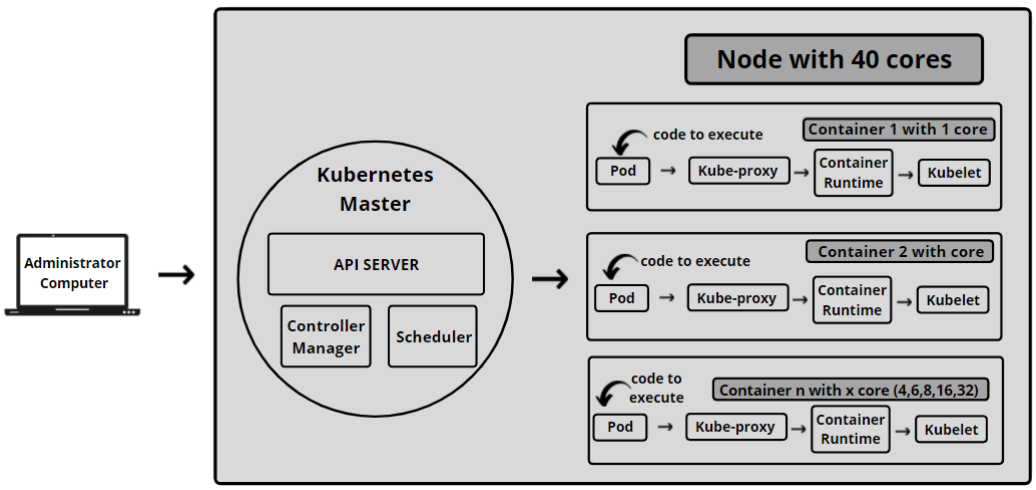


Figura 5.1: Estructura de Kubernetes implementada

La gestión del clúster de Kubernetes se realiza desde una máquina virtual externa al servidor principal, utilizando el paquete "K3sup"[1]. Esta herramienta proporciona una solución práctica para el mantenimiento y administración del clúster desde cualquier ubicación, tras una configuración inicial adecuada.

5.3. Código desarrollado

En esta sección, exploramos el código diseñado para evaluar distintas técnicas de paralelización en R, con un enfoque específico en la detección de anomalías climáticas. Se presenta inicialmente la arquitectura común del código, seguido de

implementaciones secuenciales y paralelas que utilizan un solo núcleo o varios, respectivamente.

5.3.1. Estructura común

Este segmento detalla el código base empleado en la comparativa entre diversos métodos para identificar anomalías mediante R. Se divide en la carga de librerías necesarias, la definición de una función primaria de cálculo de anomalías (Tipo I), y una función secundaria (Tipo II).

Inicialmente, se muestra la carga e instalación de las librerías esenciales 5.1. Este enfoque facilita el uso de contenedores, dado que automatiza la instalación de dependencias (por ejemplo en la línea 1) necesarias sin asumir la presencia previa de paquetes.

```
1  install.packages("dplyr")
2  library(dplyr)
3  install.packages("DDoutlier")
4  library(DDoutlier)
5  install.packages("doParallel")
6  library(doParallel)
7  install.packages("foreach")
8  library(foreach)
9  install.packages("parallel")
10 library(parallel)
```

Listing 5.1: Librerías necesarias para la ejecución correcta.

A continuación, se proporciona la primera función para el cálculo de anomalías de Tipo I 5.2, que procesa segmentos del conjunto de datos previamente filtrados por longitud y latitud, posteriormente, filtra las columnas necesarias para el análisis (línea 3) y luego aplica diversos algoritmos de detección de anomalías 4.2.

```
1  fz <- function(actual_data)
2  {
3    final_Data <- scale(actual_data[7:11], center = TRUE, scale = TRUE)
4
5    outlier_score_LOF <- LOF(final_Data, k= 5)
6    mydata_outlier <- cbind(actual_data, outlier_score_LOF)
7
8    outlier_score_LDOF <- LDOF(final_Data, k= 2)
9    mydata_outlier <- cbind(mydata_outlier, outlier_score_LDOF)
10
11   outlier_score_RDOS <- RDOS(final_Data, k = 5, h= 2)
12   mydata_outlier<- cbind(mydata_outlier, outlier_score_RDOS)
13
14   outlier_score_LOCI <- LOCI(final_Data, nn=20, k=5)$norm_MDEF
15   mydata_outlier <- cbind(mydata_outlier, outlier_score_LOCI)
16
17   outlier_score_LOOP <- LOOP(final_Data, k = 10, lambda = 3)
18   mydata_outlier <- cbind(mydata_outlier, outlier_score_LOOP)
19 }
```

Listing 5.2: Función secuencial utilizando la función tipo II.

Finalmente, la segunda función de detección de anomalías de Tipo II 5.3 se centra en trabajar con el conjunto de datos, realizando un filtrado inicial por longitud y latitud (línea 3). Esta metodología es particularmente adecuada para técnicas de paralelización que emplean clústeres para distribuir la carga de trabajo.

```
1  fz_par <- function(longitud, latitud)
2  {
3    table<-mydata[mydata$Longitud==longitud & mydata$Latitud==latitud,]
4    frame <- data.frame(table)
5    finalFrame <- data.frame(frame[7:11])
6    finalFrame <- finalFrame %>% sample_n(lengths(finalFrame), replace = FALSE)
7
8    outlier_score_LOF <- LOF(finalFrame, k= 5)
9    outlier_score_LDOF <- LDOF(finalFrame, k= 2)
10   outlier_score_RDOS <- RDOS(finalFrame, k = 5, h= 2)
11   outlier_score_LOCI <- LOCI(finalFrame, nn=20, k=5)$norm_MDEF
12   outlier_score_LOOP <- LOOP(finalFrame, k = 10, lambda = 3)
13
14   mydata_outlier <- cbind(frame, outlier_score_LOF)
15   mydata_outlier <- cbind(frame, outlier_score_LDOF)
16   mydata_outlier<- cbind(frame, outlier_score_RDOS)
17   mydata_outlier <- cbind(frame, outlier_score_LOCI)
18   mydata_outlier <- cbind(frame, outlier_score_LOOP)
19
20   mydata_outlier
21 }
```

Listing 5.3: Función secuencial utilizando la función tipo II.

5.3.2. Código secuencial

Este segmento se enfoca en el código secuencial empleado para identificar shocks climáticos, el cual se comparará posteriormente con versiones paralelizadas para determinar la viabilidad y beneficios de la paralelización en este contexto específico. Se exploran tres variantes del código: utilizando la primera función (ver Listado 5.2), la segunda función (ver Listado 5.3), y mediante un bucle "foreach" junto con la modalidad "do", aplicando la primera función (ver Listado 5.2).

En el Listado 5.4, se ilustra el uso de la primera función para procesar segmentos del dataset, previamente filtrados por latitud y longitud. Se genera un arreglo con las distintas combinaciones de estas coordenadas (línea 3), sobre el cual iteramos para aplicar la función de análisis.

```
1  sys_time_ser=0
2  sys_time_ser = system.time(
3    for (i in c(1:lengths(uniqueLongitudLatitud[1]))){
4      mydata_Ser <- filter(mydata, Longitud==uniqueLongitudLatitud[i, 1] &
       Latitud==uniqueLongitudLatitud[i, 2])
5      fz(mydata_Ser)
6    }
7  )
```

Listing 5.4: Uso de la función secuencial utilizando la función tipo I.

El Listado 5.5 muestra la aplicación de la segunda función, que requiere únicamente la longitud y la latitud como parámetros. Esta aproximación resulta más directa y eficiente en comparación con la anterior.

```
sys_time_ser_2=0
sys_time_ser_2 = system.time(
  for (i in c(1:lengths(uniqueLongitudLatitud[1]))){
    fz_par(uniqueLongitudLatitud[i, 1], uniqueLongitudLatitud[i, 2])
  }
)
```

Listing 5.5: Uso de la función secuencial utilizando la función tipo III.

En el Listado 5.6, se emplea un bucle "foreach" en combinación con la vertiente "do" para iterar sobre el conjunto de datos, usando la primera función de análisis (línea 3). Esta modalidad proporciona una alternativa interesante para evaluar frente a las otras dos implementaciones.

```
sys_time_do=0
sys_time_do=system.time(
  foreach (i=1:lengths(uniqueLongitudLatitud[1])) %do% {
    mydata_do <- filter(mydata, Longitud==uniqueLongitudLatitud[i, 1] & Latitud
    ==uniqueLongitudLatitud[i, 2])
    fz(mydata_do)
  }
)
```

Listing 5.6: Uso de el bucle "foreach" junto con la función tipo I.

Para concluir, se llevó a cabo una comparativa entre los métodos secuenciales mencionados, utilizando un dataset de datos meteorológicos con aproximadamente 18,000 registros. Los resultados, ilustrados en la Figura 5.2, indican las diferencias en tiempo de ejecución entre las variantes secuenciales examinadas, proporcionando una base para evaluar la eficacia de la paralelización en este contexto.

```
 omar@DESKTOP-J8051TQ:\$ sudo Rscript index.R
                user.self sys.self elapsed user.child sys.child
sys_time_ser          168.708    0.215 168.981      0.000     0.000
sys_time_ser_2        165.580    0.080 165.671      0.003     0.000
sys_time_do           158.671    0.120 158.791      0.000     0.000
```

Figura 5.2: Comparativa de tiempo de ejecución en métodos secuenciales

5.3.3. Código paralelo

Esta sección detalla el empleo de diversas tecnologías de paralelización examinadas anteriormente, incluyendo "foreach" con "dopar", McApply, ParSapply, y ParLapply, para optimizar la detección de shocks climáticos.

Uso de "foreach" con "dopar"

El Listado ilustra la implementación con la biblioteca "DoParallel" 5.7, utilizando la combinación de "foreach" y "dopar". Es crucial inicialmente definir el número de núcleos a utilizar (línea 1), seguido por la ejecución del bucle "foreach" con la opción "dopar" (línea 4). Aunque su implementación es directa, la limitación en la configuración representa una desventaja frente a otras tecnologías discutidas.

```
1 registerDoParallel(numCores)
2 sys_time_do_par=0
3 sys_time_do_par=system.time(
4   foreach (i=1:lengths(uniqueLongitudLatitud[1])) %dopar% {
5     mydata_do_par <- filter(mydata, Longitud==uniqueLongitudLatitud[i, 1] &
      Latitud==uniqueLongitudLatitud[i, 2])
6     fz(mydata_do_par)
7   }
8 )
```

Listing 5.7: Uso de el bucle "foreach" paralelo junto con la función tipo I.

Esta metodología destaca por su facilidad de uso y adaptabilidad a distintos contextos de análisis de datos, permitiendo una implementación eficiente y simplificada de procesos paralelos.

Implementación con McApply

El siguiente segmento explora el uso de McApply, una metodología de paralelización efectiva para procesamiento en sistemas basados en UNIX. Se caracteriza por su capacidad para dividir y ejecutar tareas en paralelo, aprovechando múltiples núcleos del procesador. Esta técnica es particularmente útil para operaciones intensivas en datos, como la detección de anomalías climáticas.

El Listado 5.8 muestra cómo McApply se emplea para la ejecución paralela de la detección de shocks climáticos. Se destaca la simplicidad en su uso: se especifican los parámetros requeridos, la función de detección de anomalías y el número de núcleos a emplear (línea 5). La función especificada debe ser capaz de procesar segmentos del conjunto de datos, en este caso, filtrando por longitud y latitud.

```
1 sys_time_mcapply=0
2 sys_time_mcapply = system.time(
3   for (i in c(1:lengths(uniqueLongitudLatitud[1]))){
4     mydata_mcapply <- filter(mydata, Longitud==uniqueLongitudLatitud[i, 1] &
      Latitud==uniqueLongitudLatitud[i, 2])
5     mclapply(uniqueLongitudLatitud[i, 1], FUN=fz_par, uniqueLongitudLatitud[i,
      2], mc.cores = numCores)
6   }
7 )
```

Listing 5.8: Uso de la tecnología "McApply" junto con la función tipo II.

Implementación con ParSapply

La metodología ParSapply representa un enfoque avanzado en el ámbito de la paralelización en R, destacando por su eficiencia en la distribución y ejecución de tareas a través de la creación de clústeres. Este enfoque es particularmente valioso cuando se manejan grandes volúmenes de datos, como en el análisis de anomalías climáticas.

El Listado 5.9 ilustra el proceso de implementación utilizando ParSapply. La secuencia comienza con la creación y configuración del clúster, especificando el número de núcleos del procesador a emplear (línea 3). Posteriormente, se procede a exportar las librerías requeridas (línea 4) y los datos necesarios para el análisis (línea 9). La ejecución paralela se lleva a cabo mediante la función ParSapply (línea 12), la cual aplica de forma distribuida la función de detección de anomalías a cada segmento de datos especificado.

```
1  sys_time_parSapply=0
2  sys_time_parSapply = system.time({
3    cl <- makeCluster(numCores)
4    clusterEvalQ(cl=cl, library(dplyr))
5    clusterEvalQ(cl=cl, library(DDoutlier))
6    clusterEvalQ(cl=cl, library(doParallel))
7    clusterEvalQ(cl=cl, library(foreach))
8    clusterEvalQ(cl=cl, library(parallel))
9    clusterExport(cl, envir = environment(), c("mydata"))
10
11   for (i in c(1:lengths(uniqueLongitudLatitud[1]))){
12     result <- parSapply(cl, uniqueLongitudLatitud[i, 1], fz_par,
       uniqueLongitudLatitud[i, 2])
13   }
14
15   stopCluster(cl)
16 })
```

Listing 5.9: Uso de la tecnología "ParSapply" junto con la función tipo II.

La implementación de ParSapply destaca por su flexibilidad y capacidad de adaptación a diferentes contextos de procesamiento de datos, ofreciendo una solución robusta para la ejecución eficiente de tareas complejas en entornos distribuidos. Este enfoque resulta especialmente útil en la investigación climática, donde el análisis detallado y la rápida obtención de resultados son esenciales.

Implementación con ParLaaply

La implementación mediante ParLapply en R representa un enfoque eficiente para la ejecución paralela de tareas, especialmente útil en análisis de gran volumen como la detección de anomalías climáticas. Este método se distingue por su capacidad para distribuir la carga de trabajo de manera efectiva a través de la creación de clústeres, optimizando así el tiempo de ejecución de los procesos.

El Listado 5.10 ilustra la implementación de ParLapply, donde se enfatiza en la preparación y configuración del clúster para su uso (línea 3). Se destaca la importancia de cargar las librerías necesarias (línea 4) y la asignación de los datos a

procesar (línea 9). La ejecución paralela se efectúa a través de la función parLapply (línea 12), aplicando de manera distribuida la función de análisis de anomalías sobre los segmentos de datos especificados por latitud y longitud.

```
1  sys_time_parLapply=0
2  sys_time_parLapply = system.time({
3    cl <- makeCluster(numCores)
4    clusterEvalQ(cl=cl, library(dplyr))
5    clusterEvalQ(cl=cl, library(DDoutlier))
6    clusterEvalQ(cl=cl, library(doParallel))
7    clusterEvalQ(cl=cl, library(foreach))
8    clusterEvalQ(cl=cl, library(parallel))
9    clusterExport(cl, envir = environment(), c("mydata"))
10
11   for (i in c(1:lengths(uniqueLongitudLatitud[1]))){
12     result <- parLapply(cl, uniqueLongitudLatitud[i, 1], fz_par,
       uniqueLongitudLatitud[i, 2])
13   }
14
15   stopCluster(cl)
16 })
```

Listing 5.10: Uso de la tecnología "ParSapply" junto con la función tipo II.

La adopción de ParLapply en proyectos de análisis climático no solo mejora significativamente los tiempos de procesamiento, sino que también facilita la gestión de recursos computacionales, permitiendo así un enfoque más dinámico y eficiente en la investigación.

Comparativa

La evaluación del desempeño de distintas estrategias de paralelización en R, especialmente en el contexto de la detección de anomalías climáticas, es crucial para optimizar los procesos de análisis de datos. La figura 5.3 presenta una comparativa entre varios métodos paralelos, utilizando un conjunto de datos meteorológicos de 18.000 registros y empleando 4 núcleos de procesamiento.

La comparativa 5.3 revela diferencias mínimas entre los métodos, lo cual sugiere que el tamaño del fichero y el número de núcleos utilizados no son suficientes para destacar las ventajas de la paralelización. Esto subraya la importancia de considerar el volumen de datos y la infraestructura de hardware al implementar técnicas de paralelización.

```
 omar@DESKTOP-J8051TQ:\$ sudo Rscript index.R
                user.self sys.self elapsed user.child sys.child
sys_time_do_par         0.022    0.100  80.347    258.981     1.049
sys_time_mcapply      160.651    0.170 160.925      0.000     0.000
sys_time_parSapply      0.068    0.010 160.376      0.002     0.000
sys_time_parLapply      0.038    0.010 160.876      0.000     0.003
```

Figura 5.3: Comparativa de tiempo de ejecución en métodos paralelos

Los resultados sugieren la necesidad de experimentar con conjuntos de datos más grandes y aumentar el número de núcleos de procesamiento para evaluar de manera efectiva el impacto de la paralelización. Además, se plantea la implementación de un clúster utilizando Kubernetes para mejorar la eficiencia de los procesos paralelos, lo cual podría proporcionar insights valiosos sobre la escalabilidad y el rendimiento de las técnicas de paralelización en análisis de datos climáticos de gran volumen.

Capítulo 6

Conclusiones y líneas futuras

6.1. Conclusiones

Este trabajo ha abordado la creciente necesidad de mejorar la detección y análisis de shocks climáticos en el contexto del cambio climático global. Mediante la integración de avanzadas técnicas de inteligencia artificial y el procesamiento de datos masivos, hemos logrado desarrollar un marco metodológico que permite identificar de manera eficaz eventos climáticos extremos. Los principales hallazgos del estudio demuestran la superioridad de ciertos algoritmos de detección de anomalías, así como la efectividad de la paralelización y la implementación con Kubernetes para optimizar el análisis de grandes volúmenes de datos climáticos.

Nuestro proyecto contribuye significativamente al campo del análisis climático, ofreciendo nuevas herramientas y metodologías para la detección precisa de shocks climáticos. Estos avances metodológicos no solo facilitan una comprensión más profunda de los patrones climáticos extremos sino que también mejoran nuestra capacidad para predecir y responder a estos eventos. Sin embargo, hemos identificado ciertas limitaciones en nuestra investigación, principalmente relacionadas con la disponibilidad y calidad de los datos climáticos, que sugieren la necesidad de futuras investigaciones para explorar fuentes de datos alternativas y mejorar los modelos de predicción climática.

Las implicaciones prácticas de nuestros hallazgos son vastas, ofreciendo aplicaciones valiosas para la gestión del riesgo climático y la formulación de políticas de adaptación y mitigación. La implementación de nuestras técnicas de detección y análisis puede ayudar a los responsables de la toma de decisiones a anticipar y prepararse mejor para los impactos del cambio climático, protegiendo así a las comunidades vulnerables y preservando los ecosistemas.

6.2. Líneas futuras

Este apartado recoge posibles direcciones para la investigación futura que podrían enriquecer y expandir los alcances del proyecto actual, ofreciendo una visión más holística sobre la detección y análisis de shocks climáticos.

En primer lugar, sería relevante investigar la eficiencia energética de las infraestructuras de procesamiento de datos utilizadas. Implementar un sistema de monitorización del consumo energético dentro de los entornos de procesamiento,

incluyendo aquellos gestionados por Kubernetes, permitiría evaluar y comparar el impacto energético de diferentes configuraciones y sistemas. Esta línea de investigación no solo busca optimizar el rendimiento computacional, sino también promover prácticas sostenibles en la investigación climática.

Además, analizar la gestión y planificación de recursos que realiza Kubernetes con los contenedores y máquinas virtuales en las infraestructuras físicas abre un campo de estudio prometedor. Identificar oportunidades de mejora en la asignación de recursos basada en las necesidades específicas de las aplicaciones de análisis climático podría resultar en un uso más eficiente y efectivo de la capacidad computacional disponible.

Otra área de interés futuro es la exploración de herramientas adicionales compatibles con la ejecución de aplicaciones en entornos contenerizados, como el NVIDIA Container Toolkit para aplicaciones que requieren GPUs. Esto es particularmente relevante para el procesamiento intensivo de datos y la ejecución de modelos de aprendizaje automático avanzados en la detección de shocks climáticos.

Investigar cómo optimizar el rendimiento de las aplicaciones mediante el modelado y análisis para determinar el número ideal de cores necesarios para su ejecución también se presenta como una línea futura de investigación valiosa. Esto implicaría un estudio detallado sobre la escalabilidad y paralelización de los procesos de análisis, buscando maximizar la eficiencia computacional.

Por último, se propone expandir el alcance del proyecto explorando la aplicación de otros tipos de análisis y procesamiento, más allá de los actualmente utilizados. Esto incluiría la incorporación de diferentes funciones analíticas y la experimentación con otros lenguajes de programación, con el fin de abordar una gama más amplia de desafíos climáticos y ambientales.

Capítulo 7

Summary and Conclusions

7.1. Conclusions

This work has addressed the growing need to improve the detection and analysis of climate shocks in the context of global climate change. Through the integration of advanced artificial intelligence techniques and massive data processing, we have managed to develop a methodological framework that allows for the effective identification of extreme climate events. The main findings of the study demonstrate the superiority of certain anomaly detection algorithms, as well as the effectiveness of parallelization and implementation with Kubernetes to optimize the analysis of large volumes of climate data.

Our project contributes significantly to the field of climate analysis, offering new tools and methodologies for the precise detection of climate shocks. These methodological advances not only facilitate a deeper understanding of extreme climate patterns but also improve our ability to predict and respond to these events. However, we have identified certain limitations in our research, mainly related to the availability and quality of climate data, which suggest the need for future investigations to explore alternative data sources and improve climate prediction models.

The practical implications of our findings are vast, offering valuable applications for climate risk management and the formulation of adaptation and mitigation policies. The implementation of our detection and analysis techniques can help decision-makers better anticipate and prepare for the impacts of climate change, thus protecting vulnerable communities and preserving ecosystems.

7.2. Future Work

This section collects possible directions for future research that could enrich and expand the scope of the current project, offering a more holistic vision of the detection and analysis of climate shocks.

Firstly, it would be relevant to investigate the energy efficiency of the data processing infrastructures used. Implementing a system to monitor energy consumption within the processing environments, including those managed by Kubernetes, would allow for the evaluation and comparison of the energy impact of different configurations and systems. This line of research not only seeks to optimize computational performance but also to promote sustainable practices in climate

research.

Furthermore, analyzing the management and planning of resources that Kubernetes performs with containers and virtual machines on physical infrastructures opens a promising field of study. Identifying opportunities for improvement in the allocation of resources based on the specific needs of climate analysis applications could result in a more efficient and effective use of available computational capacity.

Another area of future interest is the exploration of additional tools compatible with the execution of applications in containerized environments, such as the NVIDIA Container Toolkit for applications that require GPUs. This is particularly relevant for intensive data processing and the execution of advanced machine learning models in the detection of climate shocks.

Investigating how to optimize the performance of applications through modeling and analysis to determine the ideal number of cores needed for their execution also presents itself as a valuable future line of research. This would involve a detailed study of the scalability and parallelization of the analysis processes, seeking to maximize computational efficiency.

Lastly, we propose to expand the scope of the project by exploring the application of other types of analysis and processing, beyond those currently used. This would include incorporating different analytical functions and experimenting with other programming languages, with the aim of addressing a broader range of climate and environmental challenges.

Capítulo 8

Presupuesto

8.1. Justificación del presupuesto

La Tabla 8.1 muestra un resumen presupuestario del proyecto, destacando cinco áreas principales: análisis del problema, investigación de preprocesamiento, detección de shocks climáticos, evaluación del mejor algoritmo para paralelización y la investigación conclusiva.

TIPO	CANTIDAD	COSTE UNIDAD	COSTE TOTAL
Análisis de la problemática	**20 horas**		**1000 €**
Investigación Inicial	10 horas	50 €/h	500 €
Investigación sobre tecnologías a utilizar	10 horas	50 €/h	500 €
Investigación sobre el preprocesado de datos	**48 horas**		**960 €**
Reestructuración de datos	16 horas	50 €/h	800 €
Limpieza de datos	16 horas	50 €/h	800 €
Identificación de ruido	16 horas	50 €/h	800 €
Detección de shocks climáticos	**100 horas**		**5000 €**
Detección visual	24 horas	50 €/h	1200 €
Investigación algoritmos existentes	60 horas	50 €/h	3000 €
Cribado de datos	8 horas	50 €/h	400 €
Selección de la metodología de uso de los algoritmos	8 horas	50 €/h	400 €
Comparativa del mejor algoritmo para paralelizar el código	**108 horas**		**5400 €**
Investigación Inicial	24 horas	50 €/h	1200 €
Aplicación de los métodos existentes	60 horas	50 €/h	3000 €
Realización de la comparativa	24 horas	20 €/h	1200 €
Investigación Final	**64 horas**		**3200 €**
Aplicación del preprocesado de datos	16 horas	50 €/h	800 €
Aplicación de las técnicas para la detección de shocks climáticos	24 horas	50 €/h	1200 €
Muestra de resultados	24 horas	50 €/h	1200 €
TOTAL	**340 horas**		**14.600 €**

Cuadro 8.1: Presupuesto

Cada una de las áreas de investigación descritas a través de la tabla 8.1, tienen detallados una serie de subproyectos que, supondrán diferentes costes dentro del área establecida. Asimismo, debemos destacar que el proyecto ha supuesto una suma total de 14.600€ , como suma resultante de los diferentes campos del proyecto.

Bibliografía

[1] Alexellis. How do you say it? ketchup, as in tomato. https://github.com/alexellis/k3sup. Accessed: 2020-11-17.

[2] STRADATA AML. Preprocesamiento de datos: una forma de solucionar problemas antes de que aparezcan. https://aml.stradata.co/preprocesamiento-de-datos-una-forma-de-solucionar-problemas-antes-de-que-aparezc Accessed: 2021-12-08.

[3] BeServices. Kubernetes: qué es y qué aporta a nivel de programación. https://www.beservices.es/kubernetes-que-es-n-5364-es. Accessed: 2020-11-17.

[4] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, may 2000.

[5] Ismael Cabero, Irene Epifanio, Ana Piérola, and Alfredo Ballester. Archetype analysis: A new subspace outlier detection approach. *Knowledge-Based Systems*, 217:106830, 2021.

[6] Programador Clic. Reestructuración de datos de lenguaje r. https://programmerclick.com/article/7942908956/. Accessed: 2021-12-08.

[7] R Coder. sapply function in r. https://r-coder.com/sapply-function-r/. Accessed: 2021-12-20.

[8] Lázaro Cruz, Yosvany Pérez, Yudelkis Hernández, and Yaima Rodríguez. El cambio climático y sus evidencias en las precipitaciones. *Revista Cubana de Meteorología*, 22(2):3–15, 2016.

[9] DaSCI. Detección de anomalías y análisis en tiempo real, 2020.

[10] Fiego Santos (Hubspot). Data cleansing: qué es la limpieza de datos y cómo realizarla. https://blog.hubspot.es/marketing/limpieza-de-datos. Accessed: 2021-12-08.

[11] IBM. Nodo detección de anomalías. https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=models-anomaly-detection-node. Accessed: 2021-12-08.

[12] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Loop: Local outlier probabilities. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 1649–1652, New York, NY, USA, 2009. Association for Computing Machinery.

[13] Kubernetes. Production-grade container orchestration. https://kubernetes.io/. Accessed: 2021-12-23.

[14] KYOCERA. La limpieza de datos como activo empresarial. https://programmerclick.com/article/7942908956/. Accessed: 2020-12-08.

[15] Ohysucak Sciences Laboratory. University of delaware air temperature & precipitation. https://psl.noaa.gov/data/gridded/data.UDel_AirT_Precip.html#detail. Accessed: 2021-12-08.

[16] Matplotlib. Matplotlib: Visualization with python. https://matplotlib.org/. Accessed: 2021-12-08.

[17] Steve Weston Michelle Wallig, Microsoft. Package 'foreach'. https://cran.r-project.org/web/packages/doParallel/doParallel.pdf. Accessed: 2021-12-20.

[18] Microsoft. Kubernetes o docker. https://azure.microsoft.com/es-es/topic/kubernetes-vs-docker/. Accessed: 2020-11-17.

[19] Pandas. Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the python programming language. https://pandas.pydata.org/. Accessed: 2021-12-08.

[20] S. Papadimitriou, Kitagawa, H., Gibbons, P.B., Faloutsos, and C. Loci: fast outlier detection using the local correlation integral. *Proceedings 19th International Conference on Data Engineering (Cat. No.03CH37405)*,, pages 315–326, 2003.

[21] Plotly. Página oficial de plotly. https://plotly.com/. Accessed: 2021-12-19.

[22] PowerData. Calidad de datos en minería de datos a través del preprocesamiento. https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/calidad-de-datos-en-mineria-de-datos-a-traves-del-preprocesamiento. Accessed: 2021-12-08.

[23] r Jacob H. Madsen. Package 'ddoutlier'. https://cran.r-project.org/web/packages/DDoutlier/DDoutlier.pdf. Accessed: 2021-12-08.

[24] RDocumentation. clusterapply: Apply operations using clusters. https://www.rdocumentation.org/packages/parallel/versions/3.6.2/topics/clusterApply. Accessed: 2021-12-20.

[25] RDocumentation. mcapply: Apply functions over mc or mcnode objects. https://www.rdocumentation.org/packages/mc2d/versions/0.1-21/topics/mcapply. Accessed: 2021-12-20.

[26] scikit learn. Detección de valores atípicos con factor de valor atípico local (lof). https://scikit-learn.org/stable/auto_examples/neighbors/plot_lof_outlier_detection.html. Accessed: 2021-12-19.

[27] Pang-Ning Tan, Michael Steinbach, and Addison-Wesley Kumar, Vipin. *Introduction to Data Mining*. PEARSON, 1 edition, 2006.

[28] Cornell University. Explicacion ldof. https://arxiv.org/abs/0903.3257. Accessed: 2021-12-20.

[29] Juan Bosco Mendoza Vega. *R para principiantes*. ., 1st edition, 2014.

[30] Michelle Wallig, Microsoft Corporation, Steve Weston, and Dan Tenenbaum. Package 'doparallel'. https://cran.r-project.org/web/packages/doParallel/doParallel.pdf. Accessed: 2021-12-20.

[31] Hadley Wickham and Garrett Grolemund. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly Media, Inc., 1st edition, 2017.

[32] XERIDIA. La importancia del preprocesamiento de datos en inteligencia artificial: Limpieza de datosn. https://www.xeridia.com/blog/la-importancia-del-preprocesamiento-de-datos-en-inteligencia-artificial-limpieza Accessed: 2021-12-08.

[33] Cort J. Willmott y Kenji Matsuura. Precipitación y temperatura del aire terrestre: series de tiempo mensuales y anuales (1950-1999). http://climate.geog.udel.edu/~climate/html_pages/README.ghcn_ts2.html. Accessed: 2021-12-08.

[34] OMM y UNDRR. Las catástrofes relacionadas con el clima se quintuplican en 50 años, pero la mejora de los sistemas de alerta salva más vidas, 2020.

[35] Ke Zhang, Marcus Hutter, and Huidong Jin. A new local distance-based outlier detection approach for scattered real-world data. *Lecture Notes in Computer Science*, 2009.