GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

Aplicación de técnicas de aprendizaje automático para evaluar y predecir la actividad geomagnética solar en las comunicaciones

Autor: Ignacio López Soto

Director: Miguel Ángel Sanz Bobi

Julio 2022

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Aplicación de técnicas de aprendizaje automático para evaluar y predecir la actividad geomagnética solar en las comunicaciones

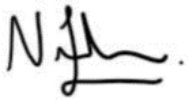
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2021/22 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.



Fdo.: Ignacio López Soto Fecha: 05/07/2022

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Firmado por SANZ BOBI MIGUEL ANGEL - ***6599** el día 06/07/2022 con un certificado emitido por AC FNMT Usuarios

Fdo.: Miguel Ángel Sanz Bobi Fecha: 06/ 07/2022

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

Aplicación de técnicas de aprendizaje automático para evaluar y predecir la actividad geomagnética solar en las comunicaciones

Autor: Ignacio López Soto

Director: Miguel Ángel Sanz Bobi

Julio 2022

Madrid

Agradecimientos

Agradecer a mi director del proyecto, Miguel Ángel Sanz Bobi, por su inestimable ayuda en la elaboración de este trabajo, por guiarme en su ejecución y por su disponibilidad durante este año.

A mis compañeros y amigos de carrera, que me han acompañado estos años.

A mi familia por su incesable apoyo.

A todos, muchas gracias.

Aplicación de técnicas de aprendizaje automático para evaluar y predecir la actividad geomagnética solar en las comunicaciones

Autor: López Soto, Ignacio

Director: Sanz Bobi, Miguel Ángel

Entidad Colaboradora: ICAI – Universidad Pontificia de Comillas

RESUMEN DEL PROYECTO

En este proyecto se entrenan y optimizan dos modelos de aprendizaje automático, uno LSTM y otro convolucional, para la predicción de valores futuros del índice *Kp*. Con la combinación de ellos, se crea una aplicación que detecta tormentas solares electromagnéticas, las cuales pueden interferir con las comunicaciones en la Tierra.

Palabras clave: LSTM, red neuronal, detección de anomalías, Kp, tormenta solar

1. Introducción

La actividad en la superficie del Sol crea un tipo de clima llamado “clima espacial”. Pese a que el Sol está a una distancia enorme de la Tierra (unos 93 millones de millas o 150 millones de kilómetros), el clima espacial puede afectar a la Tierra y al resto del sistema solar. En el peor de los casos, incluso puede dañar satélites y provocar apagones eléctricos en la Tierra. El proceso de generación de actividad geomagnética es complejo. Las tormentas geomagnéticas son las que causan los efectos señalados anteriormente en la Tierra, por lo que resulta imprescindible predecir con precisión la actividad geomagnética y su impacto en las comunicaciones. Existen índices, como el *Kp*, que miden la intensidad de las perturbaciones geomagnéticas en un cierto periodo de tiempo. El presente trabajo de fin de grado explora la posible aplicación de técnicas de aprendizaje automático para predecir el valor del indicador *Kp* y conocer así con anticipación si hay tormentas geomagnéticas que pudieran afectar a las comunicaciones terrestres.

2. Definición del proyecto

El objetivo principal del proyecto es la creación de una aplicación que permita detectar tormentas geomagnéticas a través de dos redes neuronales distintas que puedan usarse de manera conjunta para robustecer la previsión del valor de *Kp*. Con dicha aplicación se pretende identificar anomalías en la predicción del índice *Kp*. Estas anomalías corresponderán a periodos de tormenta electromagnética. Para lograr dicho objetivo, se plantean los siguientes subobjetivos que deberán conseguirse durante el desarrollo del proyecto:

- Explicar la variable *Kp* en función de otras variables disponibles, que serán las entradas de los modelos de predicción.
- Importar los datos medidos por el Space Weather Prediction Center [1] y por el Geomagnetic Observatory Niemegk, GFZ German Research Centre for Geosciences [2], que servirán como variables de entrada a las redes neuronales.

- Limpiar y procesar todos los datos recogidos de manera que se desechen o se interpolen valores de las variables que no se midieron correctamente.
- Crear y evaluar la precisión de un modelo LSTM basado en redes neuronales que trate de predecir valores futuros del índice *Kp*.
- Crear y evaluar la precisión de un modelo con una red de convolución que, al igual que el anteriormente mencionado, trate de predecir valores futuros del índice *Kp*.
- Comparar ambos modelos con el objetivo de combinarlos y mejorar la previsión de tormentas solares.
- Programar una aplicación que detecte tormentas electromagnéticas en cada día de los últimos seis años, basada en las redes neuronales entrenadas anteriormente.

3. Descripción del modelo/sistema

En primer lugar, se procedió a la importación y el tratamiento de los datos, los cuales se interpolaron en caso de no tener una entrada medida correctamente. Las variables de entrada elegidas fueron la magnitud del campo magnético, la densidad de protones en el viento solar y la velocidad de su flujo. En segundo lugar, se crearon y optimizaron el modelo LSTM y el modelo convolucional. Estos modelos son entrenados en periodos en los que no hay tormentas solares, es decir, momentos en los que *Kp* es menor de cinco. Tras varias pruebas con diferentes parámetros se llegó a los siguientes modelos, donde X representa las entradas e Y la salida del modelo:

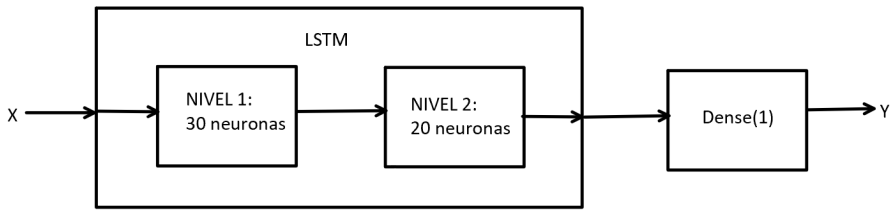


Figura 1. Arquitectura del modelo LSTM optimizado

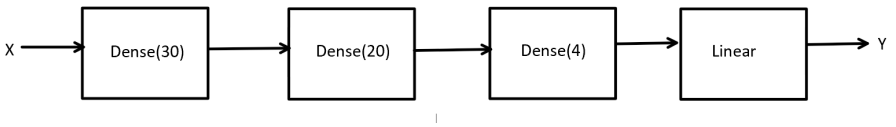


Figura 2. Arquitectura del modelo convolucional optimizado

Se puede observar que el modelo LSTM consiste en dos niveles LSTM, seguidos de una capa de activación *Dense* con una dimensión de salida. El modelo LSTM (M2) se entrenó con 1.000 épocas y un tamaño de lote de 200. Por otro lado, el modelo convolucional consta de tres capas *Dense*, cada una con el número de neuronas indicado, y una capa de activación linear. El modelo elegido (M16) se entrenó con 2.000 épocas y un tamaño de lote de 100. Para estos parámetros obtenemos los siguientes errores:

Modelos	Capas	Neuronas	Tamaño de lote	Épocas	Error Medio Training	Desviación Típica Training	Error Medio Test	Desviación Típica Test	RMSE Training	RMSE Test
M2	2	30,20	200	1000	-0.0299	0.9558	0.003	0.9627	0.9563	0.9627
M16	3	30,20,4	100	2000	-0.0025	0.8419	-0.0407	0.859	0.8419	0.86

Figura 3. Modelos elegidos y sus errores

Una vez escogidos los modelos que mejor predicen en momentos de no tormentas, se procede a comprobar su eficacia en momentos de tormentas. El objetivo es que predigan peor en

momentos de tormentas, de manera que el error en la predicción se salga del intervalo de confianza y se detecte así una anomalía. Para estudiar el acierto de las predicciones, se calcula cuántos de los errores en la predicción de periodos de tormentas solares se salen de las bandas de confianza.

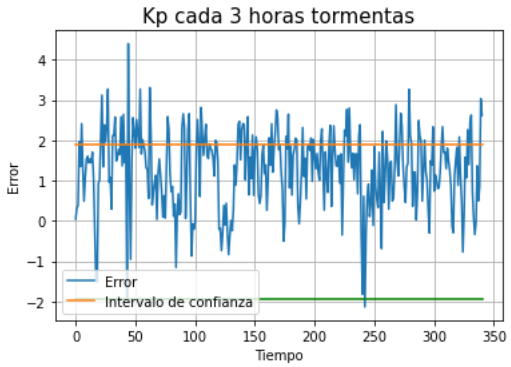


Figura 4. Error en periodo de tormentas del modelo LSTM

En la figura anterior se observa gráficamente que el 71,85% de los errores en las predicciones del modelo LSTM se mantienen dentro del intervalo de confianza.

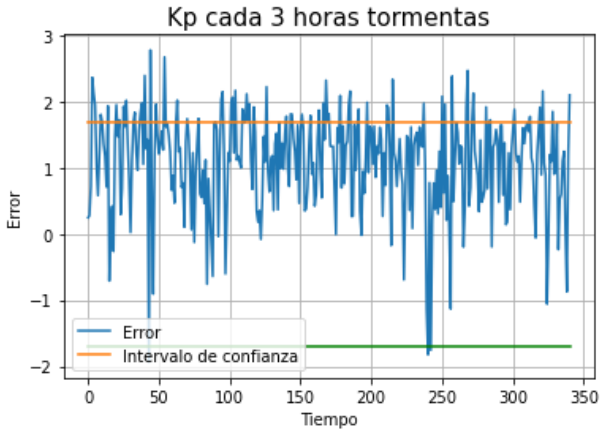


Figura 5. Error en periodo de tormentas del modelo convolucional

Por otro lado, el modelo convolucional predice el 78,29% de los datos dentro del intervalo de confianza.

Una vez escogidos los modelos, se decidió que la mejor forma de combinarlos era hacer un OR lógico de las anomalías detectadas con cada uno. El modelo resultante es el siguiente:

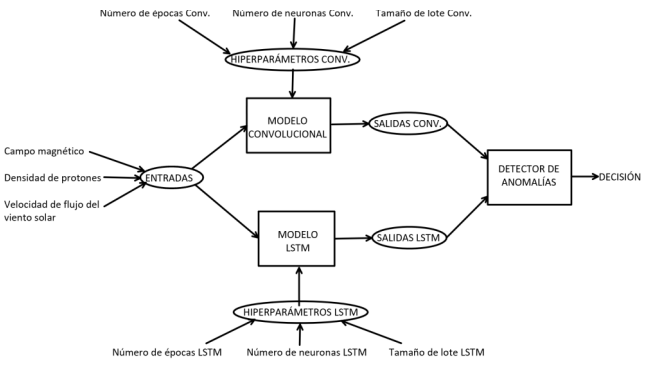


Figura 6. Esquema del detector de anomalías

Para demostrar su efectividad, podemos observar que el 62,75% de las predicciones se mantienen dentro de las bandas de confianza con este modelo.

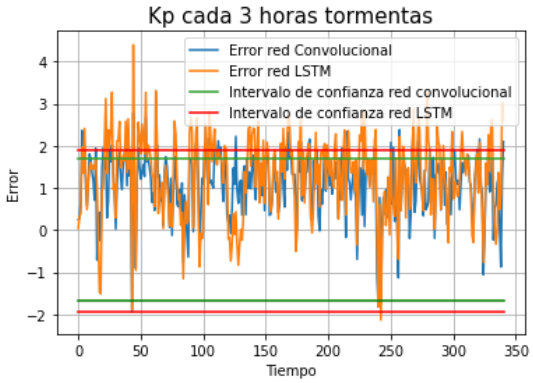


Figura 7. Errores en las predicciones de ambos modelos

Por último, se creó una aplicación que utiliza la combinación de los dos modelos para detectar tormentas cada día desde enero de 2015 hasta diciembre de 2021. La aplicación categoriza los días de verde si decide que no hubo tormenta, de amarillo si es posible que la hubiera y de rojo si está seguro de que la hubo.

4. Resultados

La aplicación consigue clasificar correctamente el 100% de los días verdes. Por lo tanto, si hay algún momento de un día en el que se mide una tormenta solar, la aplicación clasificará el mismo día de amarillo o de rojo. Sin embargo, la aplicación también clasifica erróneamente algún día en el que no hay tormentas. A continuación se muestra un ejemplo del resultado que muestra la aplicación para marzo de 2015:

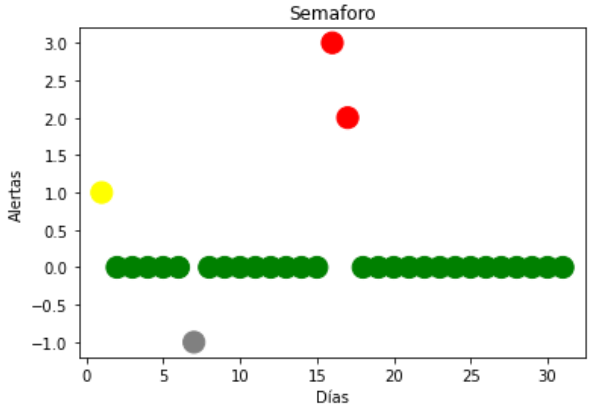


Figura 8. Resultado de la aplicación "semáforo" para marzo de 2015

Además, dentro de la aplicación se puede comprobar el funcionamiento de esta, ya que se muestran los valores reales que toma el *Kp,* además del color resultante.

	YearMonth	Day	Hours	Kp	Bt	Proton Density	Bulk Speed
593	2015 03	17	0:3	2.000	8.300000	14.100000	420.266667
594	2015 03	17	3:6	4.667	15.033333	17.433333	468.966667
595	2015 03	17	6:9	5.667	14.166667	6.266667	530.466667
596	2015 03	17	9:12	5.333	15.766667	7.000000	609.566667
597	2015 03	17	12:15	7.667	27.133333	13.700000	589.866667
598	2015 03	17	15:18	7.667	24.400000	10.066667	576.033333
599	2015 03	17	18:21	7.333	17.400000	5.033333	556.833333
600	2015 03	17	21:24	7.667	19.166667	5.766667	555.200000

Figura 9. Datos del 17 de marzo de 2015

Por ejemplo, el día 17 de marzo se muestra en rojo en la aplicación y se puede observar en la figura anterior que desde las seis de la mañana hasta el final del día hay valores de *Kp* mayores de cinco.

5. Conclusiones

En conclusión, se cumplen satisfactoriamente todos los objetivos fijados para el proyecto. Se consigue importar y tratar los datos correctamente, y se logra crear, optimizar y entrenar un modelo LSTM y un modelo convolucional. Además, se consiguen combinar ambos modelos para mejorar las previsiones; de hecho, se mejora más de un 10% el acierto de cualquiera de los dos modelos por separado. Por último, la combinación de ambos modelos se utiliza en una aplicación de creación propia para clasificar los días por color en función de si tuvo o no tuvo lugar alguna tormenta solar. La aplicación también clasifica con color amarillo o rojo la totalidad de los días con periodos de tiempo con un *Kp* superior o igual a cinco.

6. Referencias

[1] *Index of /sdb/goes/ace/monthly*. (2010). SPACE WEATHER PREDICTION CENTER. https://sohoftp.nascom.nasa.gov/sdb/goes/ace/monthly/

[2] *convenient ASCII format for Kp, ap, Ap, SN, F10.7 via FTP server*. (2021). GFZ German Research Centre for Geosciences. ftp://ftp.gfz-potsdam.de/pub/home/obs/Kp_ap_Ap_SN_F107

Application of machine learning techniques to assess and predict solar geomagnetic activity in communications

Author: López Soto, Ignacio

Supervisor: Sanz Bobi, Miguel Ángel

Collaborating entity: ICAI – Universidad Pontificia de Comillas

ABSTRACT

In this project, two machine learning models are trained and optimized. One of them is an LSTM network and the other is a convolutional network. These models try to predict future values of the *Kp* index. By combining both of them, an application is created to detect solar electromagnetic storms, which can impact communications on Earth.

Keywords: LSTM, neural network, outlier detection, *Kp*, solar storm

1. Introduction

The activity on the surface of the Sun creates a type of weather called space weather. The Sun is about 93 million miles (150 million kilometers) far from Earth. However, space weather can affect Earth and the rest of the solar system. In the worst case scenario, it can even damage satellites and cause power outages on Earth. The process of generating geomagnetic activity is complex. Geomagnetic storms are the cause of the aforementioned effects on Earth, so it is essential to accurately predict geomagnetic activity and its impact on communications. There are indices, such as *Kp*, that measure the intensity of geomagnetic disturbances over a certain period. This thesis explores the possible application of machine learning techniques to predict the value of the *Kp* indicator and, therefore, to know in advance if there are geomagnetic storms that could affect terrestrial communications.

2. Definition of the project

The main objective of the project is to create an application to detect geomagnetic storms through two different neural networks that can be used together to improve the *Kp* value forecast. The aim of this application is to detect outliers in the prediction of the *Kp* index. These outliers will correspond to electromagnetic storm periods. To achieve this objective, the following sub-objectives should be achieved during the development of the project:

- To explain the *Kp* variable in terms of other available variables that will be the inputs to the predictive models.
- To import data measured by the Space Weather Prediction Center [1] and by the Geomagnetic Observatory Niemegk, GFZ German Research Centre for Geosciences [2], which will serve as input variables to the neural networks.
- To clean and process all collected data so that the values of variables that were not measured correctly are discarded or interpolated.
- To create and evaluate the accuracy of a neural network-based LSTM model that attempts to predict future values of the *Kp* index.

- To create and evaluate the accuracy of a model with a convolutional network that, like the aforementioned model, tries to predict future values of the *Kp* index.
- To compare the two models mentioned above with the aim of combining them and improving the forecasting of solar storms.
- To program an application that detects electromagnetic storms on each day of the last six years, based on the neural networks trained as indicated above.

3. Description of the model/system

First of all, the data were imported and processed, and they were interpolated if the input was not measured correctly. The chosen input variables were magnitude of the magnetic field, proton density in the solar wind and its bulk speed. Secondly, the LSTM model and the convolutional model were created and optimized. These models are trained in periods when there are no solar storms, i.e., times when *Kp* is less than five. After several tests with different parameters, the following models were derived:

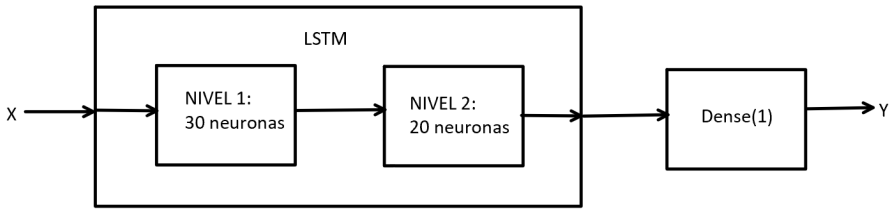


Figure 1. Optimized LSTM model architecture

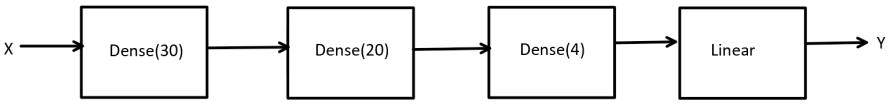


Figure 2. Optimized convolutional model architecture

It can be observed that the LSTM model consists of two layers of LSTM with a different number of neurons and an activation layer Dense, with an output dimension of 1. The LSTM model (M2) was trained with 1,000 epochs and a batch size of 200. On the other hand, the convolutional model consists of three convolutional layers, each of them with the number of neurons shown in Figure 2. Optimized convolutional model architecture, and a linear activation layer. The convolutional model (M16) was trained with 2,000 epochs and a batch size of 100:

Modelos	Capas	Neuronas	Tamaño de lote	Épocas	Error Medio Training	Desviación Típica Training	Error Medio Test	Desviación Típica Test	RMSE Training	RMSE Test
M2	2	30,20	200	1000	-0.0299	0.9558	0.003	0.9627	0.9563	0.9627
M16	3	30,20,4	100	2000	-0.0025	0.8419	-0.0407	0.859	0.8419	0.86

Figure 3. Chosen models with their errors

Once the models that best predict at non-storm times have been chosen, their accuracy is tested at storm times. The aim is for them to predict worse when there are storms, so that the prediction error is outside the confidence interval and an outlier is detected. In order to study the accuracy of the predictions, a calculation is made of how many of the errors in the prediction of solar storm periods are outside the confidence bands.

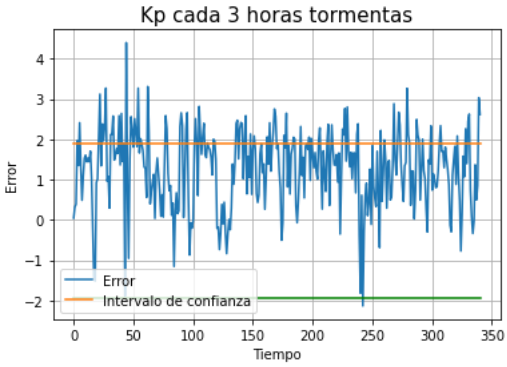


Figure 4. Prediction error in storm periods of the LSTM model

As illustrated in the figure above, 71.85% of the prediction errors of the LSTM model fall inside the confidence interval.

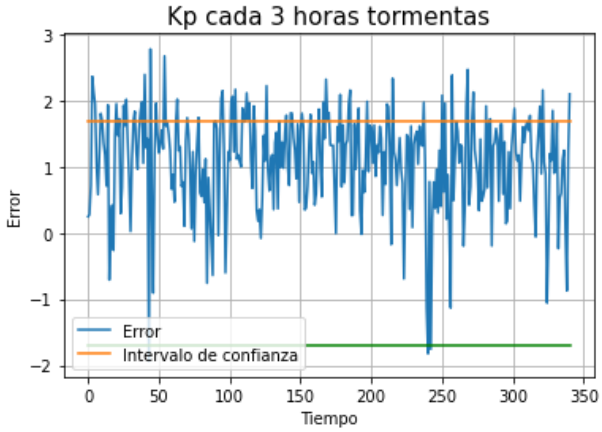


Figure 5. Prediction error in storm periods of the convolutional model

On the other hand, the convolutional model predicts 78.29% of the outputs inside the confidence interval.

Once the optimized models had been selected, it was decided that the best way of combining them was by doing a logical OR of the outliers detected by each model. The resulting model is as follows:

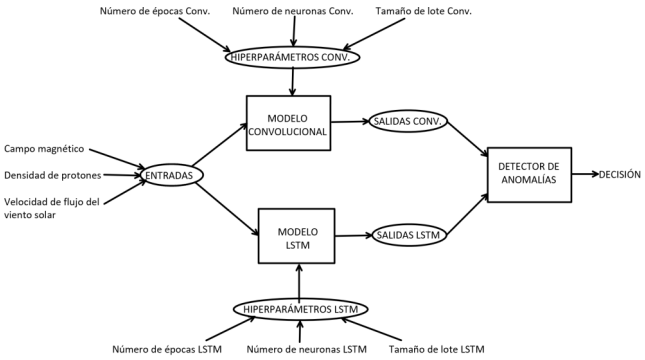


Figure 6. Scheme of the complete model

To demonstrate the effectiveness of this model, we can observe in the figure below that 62.75% of the prediction errors remain within the confidence interval.

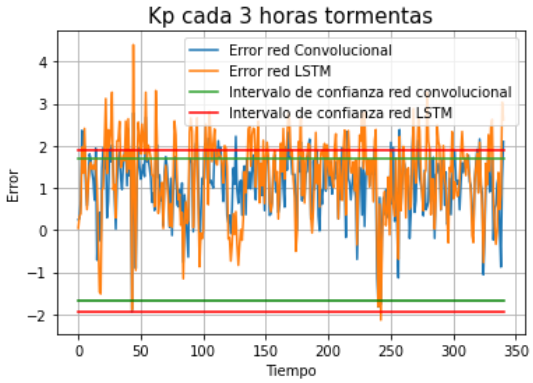


Figure 7. Prediction errors from both models

Finally, an application was created which uses the combination of both models to detect solar storms each day from January 2015 to December 2021. The application categorizes days in green if it decides that there were no storms, in yellow if there could have been a storm and in red if it is certain that there actually was a storm.

4. Results

The application correctly classifies 100% of green days. Therefore, any day on which a solar storm is detected will be classified either in yellow or in red by the application. However, some days with no storms are misclassified by the application. The following figure shows an example output of the application for March 2015:

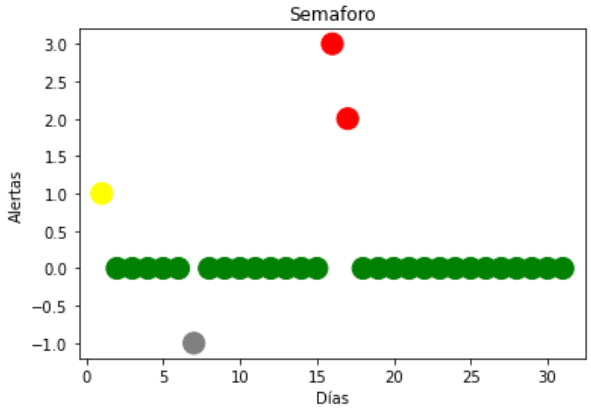


Figure 8. Results from the "traffic light" application for March 2015

Moreover, the application can be used to verify the real *Kp* values measured, and this information can be compared with the output color generated by the application to check its behavior.

	YearMonth	Day	Hours	Kp	Bt	Proton Density	Bulk Speed
593	2015 03	17	0:3	2.000	8.300000	14.100000	420.266667
594	2015 03	17	3:6	4.667	15.033333	17.433333	468.966667
595	2015 03	17	6:9	5.667	14.166667	6.266667	530.466667
596	2015 03	17	9:12	5.333	15.766667	7.000000	609.566667
597	2015 03	17	12:15	7.667	27.133333	13.700000	589.866667
598	2015 03	17	15:18	7.667	24.400000	10.066667	576.033333
599	2015 03	17	18:21	7.333	17.400000	5.033333	556.833333
600	2015 03	17	21:24	7.667	19.166667	5.766667	555.200000

Figure 9. Data From the 17th of March 2015

For example, 17 March 2015 is displayed in red in **Error! Reference source not found.**. **Error! Reference source not found.** shows that from six a.m. until the end of the day, there are measured *Kp* values greater than five.

5. Conclusions

In conclusion, all listed objectives for the project have been successfully met. The data were correctly imported and processed. Two models, one LSTM network and one convolutional network, were created, optimized and trained. Subsequently, both models were adequately combined in a way that improves predictions. In fact, the combined model improves the accuracy of either of the two separate models by more than 10%. Finally, the combination of the two models is used in a self-created application, which classifies days with different colors, depending on whether there was or was not a solar storm on that day. Moreover, the application correctly classifies every day with at least one period of solar electromagnetic storms in yellow or red.

6. References

[1] *Index of /sdb/goes/ace/monthly*. (2010). SPACE WEATHER PREDICTION CENTER. https://sohoftp.nascom.nasa.gov/sdb/goes/ace/monthly/

[2] *convenient ASCII format for Kp, ap, Ap, SN, F10.7 via FTP server*. (2021). GFZ German Research Centre for Geosciences. ftp://ftp.gfz-potsdam.de/pub/home/obs/Kp_ap_Ap_SN_F107

Índice de contenidos

1. Introducción. 1
1.1 Introducción 1
1.2 Motivación 1
1.3 Objetivos del proyecto 2
1.4 Metodología del trabajo 2
1.5 Estado de la cuestión 4
1.6 Recursos a emplear 5
2. Descripción de las tecnologías 6
2.1 Redes Neuronales Recurrentes 6
2.1.1 Red Neuronal Long Short-Term Memory (LSTM) 7
2.2 Red Neuronal Convolucional 8
3. Importación y tratamiento de datos 10
3.1 Elección de variables 10
3.2 Importación de datos 11
3.3 Tratamiento de datos 13
4. Sistema/Modelo desarrollado 15
4.1 Modelo LSTM 15
4.2 Modelo convolucional 24
4.3 Detección de anomalías 30
4.3.1 Modelo LSTM en periodo de tormentas 30
4.3.2 Modelo convolucional en periodo de tormentas 31
4.4 Elección del modelo 32
4.5 Aplicación para la detección de tormentas 34
5. Análisis de Resultados 40
6. Conclusión 42
7. Bibliografía 43
ANEXO A 45
ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS 47
ANEXO II: LISTADO Y ENLACE A LOS ARCHIVOS 48

Índice de figuras

Figura 1. Esquema de predicciones 3
Figura 2. Esquema de entradas y salida del detector de anomalías 4
Figura 3. Neurona recurrente 6
Figura 4. Neuronas recurrentes desplegadas en el tiempo 6
Figura 5. Celda de una red LSTM 7
Figura 6. Convolución en 2D [13] 8
Figura 7. Correlación entre once parámetros y Kp según el retardo [4] 10
Figura 8. Captura del archivo Kp_ap_Ap_SN_F107_2015.txt [14] 11
Figura 9. Captura del fichero 201501_ace_mag_1h.txt [15] 12
Figura 10. Captura del archivo 201502_ace_swepam_1h.txt [16] 13
Figura 11. Método de interpolación 14
Figura 12. Archivo con los datos unificados 14
Figura 13. Valores del Kp sin tormentas 15
Figura 14. Modelo LSTM 16
Figura 15. Precisión de entrenamiento en función del número de épocas LSTM 17
Figura 16. Pérdida de entrenamiento en función del número de épocas LSTM 17
Figura 17. Histograma de errores en el entrenamiento LSTM 17
Figura 18. Predicciones del Kp en entrenamiento LSTM 18
Figura 19. Histograma de errores en el test LSTM 18
Figura 20. Predicciones del Kp en test LSTM 19
Figura 21. Error medio y desviación típica en entrenamiento LSTM 19
Figura 22. Error medio y desviación típica en el test LSTM 19
Figura 23. RMSE de entrenamiento y de test LSTM 19
Figura 24. Valores reales vs valores predichos en entrenamiento LSTM 20
Figura 25. Valores reales vs valores predichos en test LSTM 20
Figura 26. Errores respecto al intervalo de confianza LSTM 20
Figura 27. Resultados para diferentes parámetros LSTM 21
Figura 28. Errores en la predicción del M5 en periodo de tormentas 22
Figura 29. Resultados M5 y M2 sin Kp en la entrada LSTM 22
Figura 30. Errores en la predicción M5 sin Kp en la entrada 23
Figura 31. Errores en la predicción M2 sin Kp en la entrada 23
Figura 32. Red convolucional 24
Figura 33. Resumen del modelo convolucional 24
Figura 34. Pérdida de entrenamiento en función del número de épocas del modelo convolucional 25
Figura 35. Histograma de errores en entrenamiento del modelo convolucional 25
Figura 36. Predicciones del Kp en entrenamiento del modelo convolucional 26
Figura 37. Histograma de errores en el test del modelo convolucional 26
Figura 38. Predicciones del Kp en test del modelo convolucional 27
Figura 39. Error medio y desviación típica del entrenamiento del modelo convolucional 27
Figura 40. Error medio y desviación típica del test del modelo convolucional 27
Figura 41. RMSE de entrenamiento y de test del modelo convolucional 27
Figura 42. Valores reales vs valores predichos entrenamiento del modelo convolucional 28

Figura 43. Valores reales vs valores predichos test del modelo convolucional 28
Figura 44. Errores respecto al intervalo de confianza del modelo convolucional 29
Figura 45. Resultados para diferentes parámetros del modelo convolucional 29
Figura 46. Errores respecto al intervalo de confianza en tormentas LSTM 31
Figura 47. Errores respecto al intervalo de confianza en tormentas modelo convolucional 32
Figura 48. Errores respecto al intervalo de confianza en tormentas OR de los modelos 33
Figura 49. Errores respecto al intervalo de confianza OR de los modelos 33
Figura 50. Menú desplegable de meses 34
Figura 51. Menú desplegable con los días del mes 35
Figura 52. Versión inicial del semáforo en marzo de 2015 36
Figura 53. Datos del 17 de marzo de 2015 36
Figura 54. Datos del 24 de marzo de 2015 37
Figura 55. Versión final del semáforo en marzo de 2015 38
Figura 56. Datos del 1 de marzo de 2015 38
Figura 57. Datos del 16 de marzo de 2015 39
Figura 58. Errores en las predicciones de ambos modelos 41

1. Introducción.

1.1 Introducción

La actividad en la superficie del Sol crea un tipo de clima llamado clima espacial. El Sol está a una distancia enorme de la Tierra: unos 93 millones de millas (150 millones de kilómetros). Sin embargo, el clima espacial puede afectar a la Tierra y al resto del sistema solar. En el peor de los casos, incluso puede dañar satélites y provocar apagones eléctricos en la Tierra. El proceso de generación de actividad geomagnética es complejo. Las tormentas geomagnéticas son las que causan el impacto señalado anteriormente en la Tierra, por lo que resulta imprescindible predecir con precisión la actividad geomagnética y su impacto en las comunicaciones. Existen índices, como el *Kp*, que miden la intensidad de las perturbaciones geomagnéticas en un cierto periodo de tiempo.

El presente trabajo de fin de grado explorará la posible aplicación de técnicas de aprendizaje automático para predecir el valor del indicador *Kp* y conocer así con anticipación si hay tormentas geomagnéticas que pudieran afectar a las comunicaciones terrestres.

1.2 Motivación

Las tormentas solares representan un riesgo para las redes de comunicaciones, que son básicas en la sociedad actual. La motivación de este trabajo es el crear una aplicación que permita detectar tormentas geomagnéticas a través de dos redes neuronales distintas que puedan usarse de manera conjunta para robustecer la previsión del valor de *Kp*. Con la creación de esta aplicación conseguiremos estudiar diferentes métodos de predicción de la actividad geomagnética solar y elegir la manera óptima de hacerlo. Este es el motivo por el que se cree conveniente la realización de este trabajo de fin de grado titulado *Aplicación de técnicas de aprendizaje automático para evaluar y predecir la actividad geomagnética solar en las comunicaciones.*

1.3 Objetivos del proyecto

El objetivo principal de este proyecto es crear una aplicación que detecte anomalías en la predicción del índice *Kp*. Estas anomalías corresponderán a periodos de tormenta electromagnética. Para lograr dicho objetivo, se plantean los siguientes subobjetivos que deberán conseguirse durante el desarrollo del proyecto:

- Explicar la variable *Kp* en función de otras disponibles que serán las entradas de los modelos de predicción.
- Importar los datos medidos por el *Space Weather Prediction Center* [1] y por el *Geomagnetic Observatory Niemegk, GFZ German Research Centre for Geosciences* [2], que servirán como variables de entrada a las redes neuronales.
- Limpiar y procesar todos los datos recogidos de manera que se desechen o se interpolen valores de las variables que no se midieron correctamente.
- Crear y evaluar la precisión de un modelo LSTM basado en redes neuronales que trate de predecir valores futuros del índice *Kp*.
- Crear y evaluar la precisión de un modelo con una red de convolución que, al igual que el anteriormente mencionado, trate de predecir valores futuros del índice *Kp*.
- Comparar ambos modelos mencionados con el objetivo de combinarlos y mejorar la previsión de tormentas solares.
- Programar una aplicación que detecte tormentas electromagnéticas en cada día de los últimos seis años, basada en las redes neuronales entrenadas anteriormente.

1.4 Metodología del trabajo

En primer lugar, se eligen las variables de entrada a los modelos en función de su correlación con la variable *Kp*. Estas son el campo magnético, la densidad de protones, la velocidad de propagación de los protones y el propio *Kp*.

Una vez elegidas, se procede a la importación de estos datos de repositorios públicos. Existen casos en los que se realizan mediciones incorrectas o no se registra ninguna medida. Para contar con el mayor número de datos, en la medida de lo posible, se suplirán estas entradas con valores interpolados

linealmente a partir de los valores anteriores y posteriores. En casos en los que fallan más de dos medidas seguidas, simplemente se han desechado esas entradas. Posteriormente, se juntarán todos estos datos en el mismo archivo, ordenados por fecha e intervalo de tres horas. Concretamente, se trabajará con datos medidos cada día desde enero de 2015 hasta diciembre de 2021.

A continuación, se crean y entrenan los modelos de predicción. En ambos casos, se entrenan las redes con entradas correspondientes a periodos en los que no hubiera tormenta electromagnética, es decir, momentos en los que el valor del *Kp* sea menor de 5. Para ello, se reúnen todos los datos pertenecientes a estos periodos y se utilizan aproximadamente los primeros dos tercios de ellos, correspondientes a medidas tomadas más atrás en el tiempo, para el conjunto de entrenamiento, y el último tercio, correspondiente a las medidas más recientes, para el conjunto de test. A través de distintas gráficas, se evalúan estos modelos y se intentan optimizar al máximo en función de sus parámetros. Estos parámetros son el número de épocas de entrenamiento del modelo, el tamaño de lote y el número de neuronas que se asigna a cada capa. En la siguiente figura (Figura 1), se puede observar gráficamente que las predicciones, o salidas, del modelo que usemos, dependen de las entradas y de los valores de los hiperparámetros que escojamos. En concreto, las salidas serán valores de *Kp* correspondientes a las tres horas siguientes a la medición de las entradas. El bloque "MODELO" representa abstractamente la arquitectura de la red neuronal que elijamos.

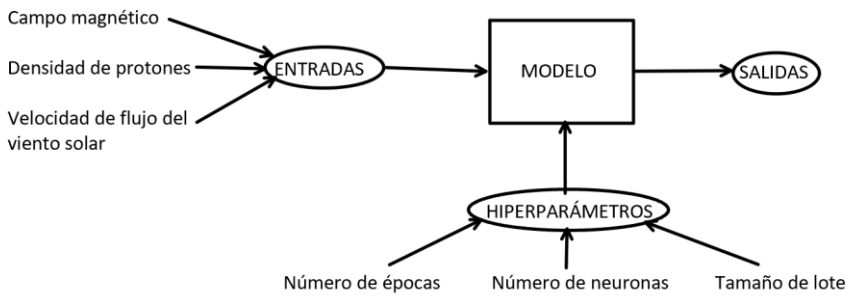


Figura 1. Esquema de predicciones

Para decidir qué modelo es el mejor, nos basamos en varios indicadores. En primer lugar, el error medio de las predicciones en el test debe ser lo más cercano a cero posible. Además, la desviación típica de estos errores debe ser lo más pequeña que se pueda. En segundo lugar, se comparará el error cuadrático medio del test, que también debe ser lo menor posible. Finalmente, observaremos la eficacia de los modelos a la hora de detectar anomalías midiendo las que detectan en función de las que debería detectar. Una vez encontrado el mejor modelo, se procede a utilizarlo para predecir valores del *Kp* correspondientes a periodos en los que sí tuvieran lugar tormentas electromagnéticas. Dado que los modelos están entrenados en periodos en los que no hay tormentas, deberían predecir erróneamente en

periodos en los que sí las haya. Para medir esto, designamos unas bandas de confianza del error que tiene la predicción respecto a los datos originales. El objetivo es que dicho error se encuentre dentro de las bandas en periodos en los que no hay tormentas y fuera de ellas en periodos en los que sí las hay.

Finalmente, después de optimizar ambos modelos, se procede a elegir la mejor forma de predecir el *Kp* de manera que haya el mayor número de anomalías posible en periodos de tormentas. Una vez hecho esto, se programa una aplicación interactiva que muestre gráficamente la actividad geomagnética en ciertos periodos de tiempo.

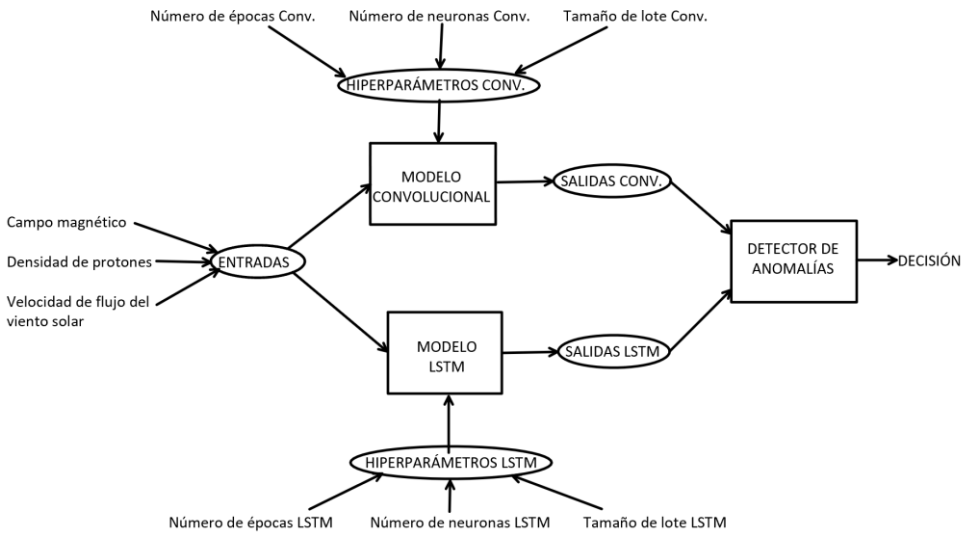


Figura 2. Esquema de entradas y salida del detector de anomalías

En la figura anterior (Figura 2), se muestra gráficamente el proceso de detección de anomalías. En primer lugar, se introducen las entradas a los modelos, donde "Conv." hace referencia al modelo convolucional. Con las entradas y los hiperparámetros, cada modelo obtiene sus respectivas predicciones y haciendo uso de ambas, la aplicación decide si ha tenido lugar o no una anomalía.

1.5 Estado de la cuestión

El actual estado del arte de la predicción del índice *Kp* hace uso de diferentes métodos como funciones derivadas empíricamente, modelos basados en la física y redes neuronales. La mayor parte de trabajos relacionados con la predicción de la actividad geomagnética solar hace uso de redes neuronales recurrentes para realizar pronósticos de valores futuros del índice *Kp*. En concreto, varios trabajos como los referenciados en [3], [4] y [5] emplean una red LSTM (*Long Short-Term Memory)*, un tipo de red neuronal recurrente. En este trabajo, se utilizará una red LSTM, además de una red convolucional, cuyo uso he encontrado menos común.

Sin embargo, a diferencia del objetivo general de los trabajos que se centran en la predicción de tormentas solares, este trabajo no pretende predecir correctamente los valores de *Kp* correspondientes a tormentas solares, es decir, valores de *Kp* mayores de cinco. Lo que pretende hacer este trabajo, es predecir correctamente periodos en los que no hay tormentas, de manera que cuando el error en la predicción supere una cierta banda de confianza, se detecte una anomalía que sea causada por un periodo de tormenta. En otras palabras, no se busca predecir valores futuros de *Kp* que no estén medidos, si no que se pretende utilizar técnicas de aprendizaje automático para evaluar la actividad geomagnética solar.

Otro método de predicción del índice *Kp* que siguen algunos trabajos como [4] y [5] es dividir la predicción en dos partes. En primer lugar, se utiliza un modelo que predice si habrá tormenta o no. En segundo lugar, para predecir el valor del *Kp* se usa un modelo distinto en cada caso. En este trabajo, se empleará el mismo modelo para predecir valores de tormentas y de no tormentas.

Los artículos citados consiguen un buen comportamiento por parte de sus modelos y afirman que el uso de redes neuronales LSTM es realmente un buen método a la hora de predecir el valor del índice *Kp*.

1.6 Recursos a emplear

Para llevar a cabo el proyecto, se trabajará en el entorno Anaconda. Dentro de Anaconda, trabajaremos con Jupyter Notebooks versión 6.4.5 [6], un entorno de desarrollo de código, en este caso en el lenguaje de programación Python, versión 3.9.7 [7]. Además, para el desarrollo de las redes neuronales se empleará keras RNN [8], un API que facilita la utilización y la customización de redes neuronales recursivas (RNN). También se hará uso de las librerías matplotlib, versión 3.4.3 [9], para la representación de gráficos, de pandas, versión 1.3.4 [10], para la lectura y organización de la estructura de los datos; y de ipywidgets, versión 7.6.5 [11], para facilitar un entorno gráfico amigable en el detector de anomalías. Además, se usarán las librerías numpy, versión 1.20.3 [12], para realizar cálculos con vectores.

Cabe destacar que el proyecto se podía desarrollar en Matlab o en R, además de en Python. Sin embargo, se decidió utilizar Python debido a su comodidad a la hora de programar, a pesar de la poca experiencia que tenía con el lenguaje.

2. Descripción de las tecnologías

2.1 Redes Neuronales Recurrentes

Las redes neuronales recurrentes son un tipo de red neuronal que analizan datos de series temporales, de manera que se tiene en cuenta la dimensión temporal. Esta clase de redes fueron concebidas en la década de 1980, pero no era posible llevarlas a la práctica por la falta de potencia de computación. Como se puede intuir, una red neuronal recurrente, está formada por neuronas recurrentes. Estas se pueden representar de la siguiente manera:

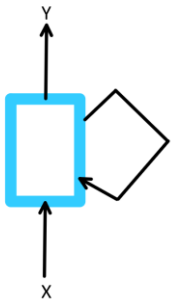


Figura 3. Neurona recurrente

En la Figura 3, X representa la entrada e Y la salida. Además, la neurona utiliza su propia salida del instante de tiempo anterior para calcular junto con la entrada, la siguiente salida. Si desplegáramos esta neurona en el eje del tiempo, obtendríamos la siguiente representación:

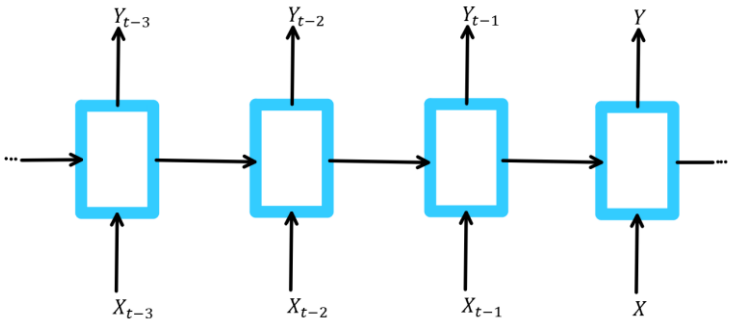


Figura 4. Neuronas recurrentes desplegadas en el tiempo

Por lo tanto, a diferencia de otras redes neuronales, la salida en una red neuronal recurrente se calcula a partir de dos conjuntos de parámetros. El primer conjunto corresponde a la entrada (X) y el segundo a la salida del instante anterior. Debido a la dependencia que guardan las futuras salidas con respecto a las anteriores, es necesario tener algún tipo de memoria que almacene esa información. En

concreto, esta parte de las redes neuronales recurrentes que se encarga de almacenar la memoria se llama *memory cell*, célula de memoria en inglés. Gracias a estas células, conseguimos aplicar este tipo de redes neuronales a cosas como el procesamiento de lenguaje natural, el procesamiento de imágenes y el procesamiento de vídeos.

2.1.1 Red Neuronal Long Short-Term Memory (LSTM)

Las redes neuronales *Long Short-Term Memory* son una extensión de las redes neuronales recurrentes en la que se capturan mejor las dependencias a largo plazo. En concreto una celda de una red neuronal recurrente LSTM se puede esquematizar de la siguiente forma:

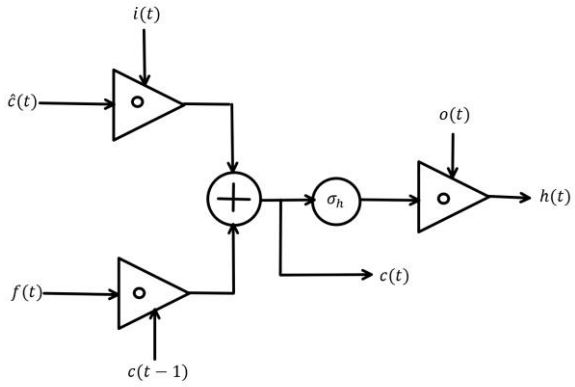


Figura 5. Celda de una red LSTM

En el diagrama, i(t) (*input gate*) se encarga de decidir si la entrada en t es importante o no. Por otro lado, f(t) (*forget gate*) está encargado de decidir si c(t-1) debería olvidarse o no. Finalmente, o(t) (*output gate*) decide qué cantidad de c(t) debería estar expuesta. En definitiva, las variables siguen las siguientes fórmulas:

$$i(t) = \sigma_i(W_{ix}x(t) + W_{ih}h(t-1) + W_{ib})$$

Ecuación 1. Input gate

$$f(t) = \sigma_f(W_{fx}x(t) + W_{fh}h(t-1) + W_{fb})$$

Ecuación 2. Forget gate

$$o(t) = \sigma_o(W_{ox}x(t) + W_{oh}h(t-1) + W_{ob})$$

Ecuación 3. Output gate

$$\hat{c}(t) = \sigma_{\hat{c}}(W_{\hat{c}x}x(t) + W_{\hat{c}h}h(t-1) + W_{\hat{c}b})$$

Ecuación 4. Nueva célula de memoria

$$c(t) = f(t)\ o\ c(t-1) + i(t) o\ \hat{c}(t)$$

Ecuación 5. Célula de memoria final

$$h(t) = o(t)\ o\ \sigma_{\mathrm{h}}(c(t))$$

Ecuación 6. Memoria oculta

En las ecuaciones, sigma representa funciones de activación, o representa un producto de *Hadamard* y el significado de cada variable que se incluye en la Figura 5 está brevemente descrito en los títulos de las ecuaciones. Además, "W" representa los pesos (*weights* en inglés) de cada variable. Este tipo de red neuronal recurrente resuelve algunos problemas a los que se enfrentan las redes neuronales recurrentes originales, como son la dependencia a largo plazo y el problema de desvanecimiento del gradiente.

2.2 Red Neuronal Convolucional

Las redes convolucionales son un tipo de red neuronal en el que las neuronas corresponden a campos receptivos de maneta similar a las neuronas en la corteza visual primaria de un cerebro biológico. Este tipo de redes, consisten en múltiples capas de filtros convolucionales de una o más dimensiones. Generalmente, se añade una función de activación después de filtrar. Una sola capa convolucional se compone de una matriz de entrada, un filtro y una matriz de salida.

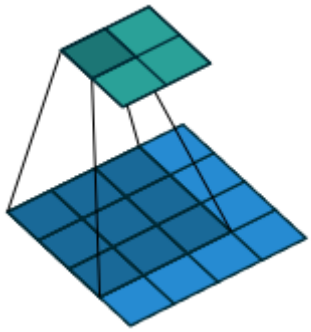


Figura 6. Convolución en 2D [13]

En la imagen, la matriz azul representa la entrada, la matriz azul oscura el filtro y la matriz verde la salida. Un ejemplo del funcionamiento de una red neuronal convolucional es el siguiente:

$$\begin{pmatrix} 1 & -1 & 0 \\ 2 & 1 & 3 \\ -1 & 0 & 2 \end{pmatrix}$$

Entrada a la red convolucional

$$\begin{pmatrix} 1 & 0 \\ 2 & -1 \end{pmatrix}$$

Filtro de la red convolucional

$$\begin{pmatrix} 4 & -2 \\ 0 & -1 \end{pmatrix}$$

Salida de la red convolucional

La primera matriz sería la entrada al modelo, la segunda el filtro, y la tercera el resultado de aplicar el filtro a la entrada. El resultado se obtiene sumando los productos entre el filtro y la entrada varias veces. En concreto, el cuatro de la salida, se obtiene de realizar la suma $(1 * 1) + (-1 * 0) + (2 * 2) + (1 * -1)$. Aplicando el filtro a toda la entrada, deslizándolo como se ilustra en la Figura 6**Error! Reference source not found.**, obtenemos todos los valores de la salida. La red neuronal convolucional se encarga de encontrar los valores del filtro que más ajustan las salidas a los valores que debería predecir.

3. Importación y tratamiento de datos

En esta sección se exponen los pasos iniciales de este proyecto. En primero lugar, se estudia qué variables guardan una mayor correlación con la variable *Kp*. Después, se importan datos de todas las variables seleccionadas, organizadas por fecha y con entradas cada periodo de tres horas. Por último, se limpian las entradas erróneas de estos datos y se juntan todas en varios archivos.

3.1 Elección de variables

Basándonos en las variables disponibles relacionadas con la actividad geomagnética, consideramos cuatro parámetros candidatos a ser entradas de los modelos de predicción. Estos parámetros son, respecto al campo magnético interplanetario (IMF) la magnitud total del mismo, y respecto al viento solar, la densidad de protones y la velocidad de su flujo. Por último, tendremos en cuenta también el propio *Kp*. Basándonos en los estudios [4] y [5], y por simplicidad a la hora de la extracción de los datos, usaremos todas las variables mencionadas ya que guardan cierta correlación con *Kp*. En el siguiente gráfico extraído de [4], se puede observar la correlación del *Kp* con distintas variables:

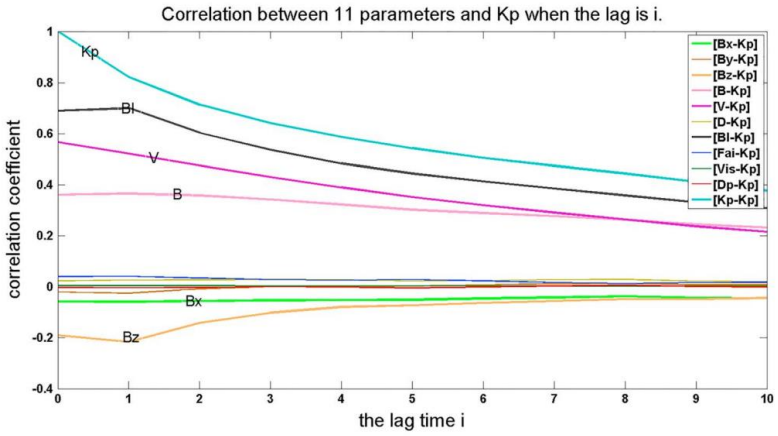


Figura 7. Correlación entre once parámetros y Kp según el retardo [4]

En la Figura 7 se muestra que el *Kp* guarda cierta correlación con el campo magnético (B), la densidad de protones(D) y la velocidad de su flujo(V). La variable *Bl*, es el *Boyle Index*, y a pesar de

guardar la mayor correlación con *Kp*, no se ha elegido como variable de entrada porque es un índice que se calcula a partir de otros con una fórmula matemática. Por otro lado, las entradas elegidas se miden directamente y se pueden obtener con facilidad.

3.2 Importación de datos

En este apartado se expone cómo se han importado los datos de las variables elegidas. Los datos se obtienen de dos fuentes distintas. Por un lado, los valores del índice *Kp* están recogidos y disponibles al público por GFZ *German Research Centre for Geosciences* [2]. Por otro lado, los datos respectivos al viento solar y al campo magnético interplanetario se han recogido del *Space Weather Prediction Center: National Oceanic and Atmospheric Administration* [1].

Se recogen los datos de tres tipos de archivos diferentes:

1. Índice *Kp*: Se descargan archivos correspondientes a cada año desde 2015 hasta 2021 de un servidor FTP [2]. De estos archivos, se extraen ocho valores del *Kp* por día, cada uno correspondiente a un periodo de tres horas. A través de un *script* sencillo de Python[1], se automatiza este proceso y cada valor del *Kp* se guarda en un archivo asociado a su fecha e intervalo de horas. Los ficheros tienen el siguiente aspecto:

```
# The format for each line is (i stands for integer, f for float):
#iii ii ii iiiii fffff.f iiii ii ff.fff ff.fff ff.fff ff.fff ff.fff ff.fff ff.fff ff.fff :
# The parameters in each line are:
#YYY MM DD  days  days_m  Bsr dB    Kp1    Kp2    Kp3    Kp4    Kp5    Kp6    Kp7    Kp8
2015 01 01 30316 30316.5 2475  5  2.333  1.000  1.000  1.000  1.333  1.333  1.667  2.333
2015 01 02 30317 30317.5 2475  6  3.000  1.000  1.333  1.667  2.333  2.333  3.000  4.000
2015 01 03 30318 30318.5 2475  7  4.667  3.000  2.333  3.000  2.667  1.000  1.333  0.000
2015 01 04 30319 30319.5 2475  8  0.667  1.000  2.000  3.333  3.667  5.000  3.333  4.000
2015 01 05 30320 30320.5 2475  9  5.333  3.000  2.333  1.667  2.000  1.667  2.333  3.000
```

Figura 8. Captura del archivo Kp_ap_Ap_SN_F107_2015.txt [14]

2. Magnitud total del campo magnético (*Bt*): Se descargan archivos correspondientes a cada mes desde enero de 2015 hasta diciembre de 2021. Los archivos son descargados desde un repositorio de la NASA [1]. En estos archivos, conseguimos el campo magnético correspondiente a cada hora de cada día del mes; por lo tanto, para conseguir unos datos consistentes con los del índice *Kp*, realizo la media los valores recogidos

[1] Ver KpDataConverter.ipynb en el Anexo II

cada tres horas a través de un *script* también en Python[2], que guarda en un archivo la media de cada tres horas junto a su fecha e intervalo de horas correspondientes. Además, al realizar la media se han tenido en cuenta los datos que estaban mal medidos y para los que no se había recogido una entrada. Para ello, las medias resultan de la media aritmética entre los valores disponibles, o -1 en caso de no existir datos para ninguna de las tres horas. El aspecto de los archivos de los que obtenemos los datos es el siguiente:

```
#                  Modified Seconds
# UT Date    Time  Julian   of the   ----------------  GSM Coordinates ---------------
# YR MO DA   HHMM    Day      Day     S      Bx       By       Bz       Bt      Lat.    Long.
#-------------------------------------------------------------------------------------------
2015 01 01   0000   57023        0     0     4.2     -2.2      0.8      4.8      9.1    332.1
2015 01 01   0100   57023     3600     0     3.2     -2.5     -0.6      4.1     -8.2    322.3
2015 01 01   0200   57023     7200     0     2.3     -2.9      0.7      3.8     10.9    307.7
2015 01 01   0300   57023    10800     0     1.7     -2.5     -0.6      3.1    -11.3    304.1
2015 01 01   0400   57023    14400     0     2.6     -2.1     -0.2      3.3     -3.6    321.0
2015 01 01   0500   57023    18000     0     2.2     -2.1     -0.1      3.1     -2.3    316.3
2015 01 01   0600   57023    21600     0     1.9     -1.5      0.3      2.5      5.9    321.5
2015 01 01   0700   57023    25200     0    -2.6     -0.9     -1.1      3.0    -20.8    198.7
2015 01 01   0800   57023    28800     0    -2.8     -1.9     -0.5      3.4     -8.1    213.9
2015 01 01   0900   57023    32400     0    -0.3     -0.6      0.3      0.7     26.8    241.3
2015 01 01   1000   57023    36000     0     3.8     -0.6     -0.5      3.9     -7.2    350.9
2015 01 01   1100   57023    39600     0     1.0     -1.0      0.1      1.4      4.4    314.0
2015 01 01   1200   57023    43200     0    -0.9     -2.0      0.6      2.2     14.2    245.6
2015 01 01   1300   57023    46800     0     0.3     -1.8      1.0      2.1     29.0    279.2
2015 01 01   1400   57023    50400     0    -2.6     -1.5     -0.1      3.0     -2.6    210.0
2015 01 01   1500   57023    54000     0    -4.6     -1.8     -0.1      4.9     -1.1    201.7
2015 01 01   1600   57023    57600     0    -4.7     -2.1     -0.6      5.2     -6.8    204.6
2015 01 01   1700   57023    61200     0    -4.8     -2.2     -1.1      5.3    -11.6    204.5
2015 01 01   1800   57023    64800     0    -2.3     -2.3     -1.6      3.6    -25.8    224.5
```

Figura 9. Captura del fichero 201501_ace_mag_1h.txt [15]

3. Densidad de protones y velocidad del flujo del viento solar: Se descargan los archivos correspondientes a cada mes desde enero de 2015 hasta diciembre de 2021. Los archivos son descargados del mismo repositorio que en el punto anterior. En estos archivos conseguimos datos correspondientes a las dos variables para cada hora de cada día del mes del archivo. De la misma manera que con el campo magnético, se utiliza un *script* en Python[3] con el que se leen los archivos y se guardan las medias de cada tres horas de los datos de la manera indicada en el punto anterior. El aspecto de los archivos de los cuales obtenemos los datos es el siguiente:

[2] Ver MagenticFieldDataConverter.ipynb en el Anexo II
[3] Ver solarWindDataConverter.ipynb en el Anexo II

```
#               Modified Seconds   -------------  Solar Wind  -----------
# UT Date   Time  Julian  of the          Proton      Bulk         Ion
# YR MO DA  HHMM    Day     Day     S    Density     Speed     Temperature
#-------------------------------------------------------------------------
2015 01 01  0000   57023       0    0        2.1      555.8     1.40e+05
2015 01 01  0100   57023    3600    1        2.2      575.3     1.42e+05
2015 01 01  0200   57023    7200    0        1.9      557.3     1.10e+05
2015 01 01  0300   57023   10800    0        2.1      550.6     1.21e+05
2015 01 01  0400   57023   14400    0        1.7      534.5     1.32e+05
2015 01 01  0500   57023   18000    0        1.7      507.2     1.21e+05
2015 01 01  0600   57023   21600    0        2.0      523.5     1.48e+05
2015 01 01  0700   57023   25200    0        1.9      538.6     1.37e+05
2015 01 01  0800   57023   28800    0        2.1      521.0     1.30e+05
2015 01 01  0900   57023   32400    0        2.0      501.6     1.20e+05
2015 01 01  1000   57023   36000    0        1.9      479.9     1.43e+05
2015 01 01  1100   57023   39600    0        2.9      474.1     9.34e+04
2015 01 01  1200   57023   43200    0        4.1      449.0     4.96e+04
2015 01 01  1300   57023   46800    0        4.4      442.4     5.07e+04
2015 01 01  1400   57023   50400    0        4.1      448.0     5.57e+04
2015 01 01  1500   57023   54000    0        3.9      456.3     3.95e+04
2015 01 01  1600   57023   57600    0        4.1      448.9     4.20e+04
```

Figura 10. Captura del archivo 201502_ace_swepam_1h.txt [16]

3.3 Tratamiento de datos

En este apartado se explica cómo se han tratado los datos una vez obtenidos. Una vez conseguidos y organizados los datos de las distintas variables, lo siguiente es unificarlos en un solo archivo. Para ello, se hace uso de otro script en Python[4]. En él, se leen los tres archivos que conseguimos anteriormente y se juntan en tres archivos diferentes. En el primero, se recogen los datos correspondientes a periodos en los que no ocurren tormentas. En el segundo, se recogen datos correspondientes a periodos en los que sí tienen lugar tormentas solares. En el último, se recogen todos los datos ordenados cronológicamente.

A la hora de unificar los datos en los distintos archivos, se tiene en cuenta que hay datos que están mal medidos o que tienen la entrada vacía. Para ello, se rellenan estos datos por medio de interpolación lineal de la siguiente forma:

[4] Ver DataUnifyer.ipynb en el Anexo II

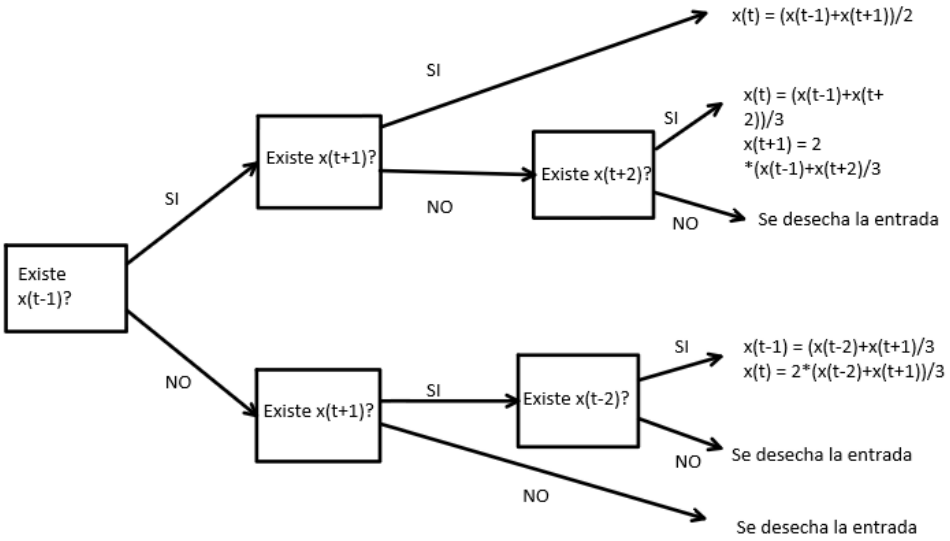


Figura 11. Método de interpolación

Finalmente, una vez tratadas todas esas entradas, se guardan en los archivos correspondientes en función de su valor de *Kp*. Si el *Kp* es menor que cinco, entonces esa entrada corresponde a un periodo en el que no hay tormenta. En el caso contrario, corresponde a un periodo en el que sí hay tormenta. El resultado obtenido es el siguiente:

	A	B	C	D	E	F	G
1	YearMonth	Day	Hours	Kp	Bt	Proton Density	Bulk Speed
2	2015 01	1	00:03	2.333	4.233333333	2.066666667	562.8
3	2015 01	1	03:06	1	3.166666667	1.833333333	530.7666667
4	2015 01	1	06:09	1	2.966666667	2	527.7
5	2015 01	1	09:12	1	2	2.266666667	485.2
6	2015 01	1	12:15	1.333	2.433333333	4.2	446.4666667
7	2015 01	1	15:18	1.333	5.133333333	4.1	450.5333333
8	2015 01	1	18:21	1.667	3.166666667	5.1	435.6
9	2015 01	1	21:24	2.333	3.866666667	4.733333333	436.7333333
10	2015 01	2	00:03	3	4.133333333	5.033333333	432.1
11	2015 01	2	03:06	1	4.6	5.6	420.9
12	2015 01	2	06:09	1.333	5.2	9.1	430.8
13	2015 01	2	09:12	1.667	6.933333333	5.1	443.8333333
14	2015 01	2	12:15	2.333	6.733333333	7.466666667	421.4333333

Figura 12. Archivo con los datos unificados

4. Sistema/Modelo desarrollado

En este apartado se explican los pasos que se han seguido para construir los modelos de comportamiento y el detector de tormentas. En los apartados 4.1 y 4.2 se exponen las consideraciones que se han seguido para crear cada tipo de modelo, aunque ambos comparten algunos parámetros y las variables de entrada.

4.1 Modelo LSTM

El primer paso para la creación del modelo LSTM es la creación de un conjunto de datos de entrenamiento y un conjunto de datos de test. Para ello, leemos el archivo con los datos unificados sin tormentas y pasamos los datos a un *dataframe* de la librería *pandas* [10]. Esta tabla de datos consiste en valores del campo magnético, de la densidad de protones y de la velocidad de flujo del viento solar como entradas. Con estas entradas se intentará predecir el valor de *Kp* de las próximas tres horas como salida.

Lo siguiente es normalizar esos datos, lo cual se ha hecho haciendo uso del *MinMaxScaler* [17] de la librería *sklearn* [18]. Una vez normalizados, los separamos en conjunto de entrenamiento y conjunto de test. Para ello, de las 19.442 entradas, asignamos un poco más de dos tercios de los datos al conjunto de entrenamiento. Estos son los valores del *Kp* en cada instante:

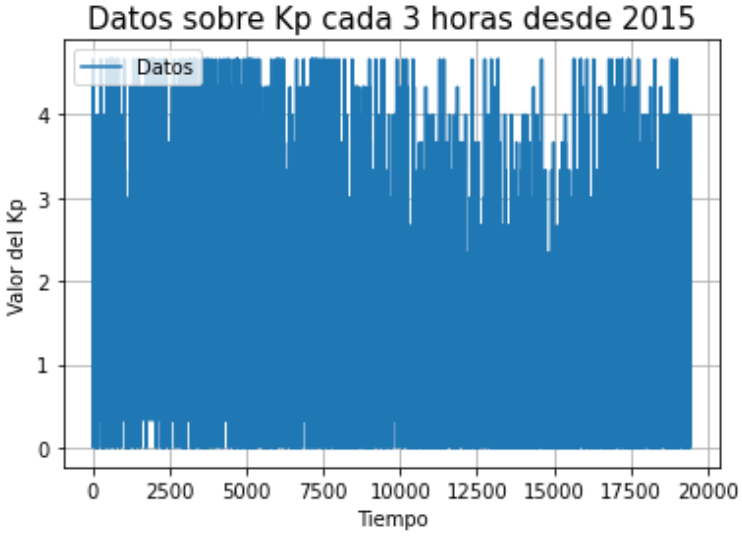


Figura 13. Valores del Kp sin tormentas

Normalmente, se escogerían simplemente los primeros dos tercios de los datos como conjunto de entrenamiento. Sin embargo, debido a la forma de los datos, los dos primeros tercios (hasta el dato 12.831) son bastante regulares, en el sentido de que no hay muchas oscilaciones en los valores del *Kp*. Por otro lado, en el último tercio de los datos existen mayores oscilaciones de estos valores. Con el objetivo de tener mejor entrenado el modelo, es decir, entrenado con periodos de más y de menos oscilaciones en la salida, se amplía el tamaño del conjunto de entrenamiento hasta la entrada número 15.602.

El siguiente paso es la creación de la red en sí. El modelo es relativamente sencillo y consiste en dos capas de LSTM con distinto número de neuronas cada una y una capa de activación al final. El modelo se puede representar de la siguiente forma:

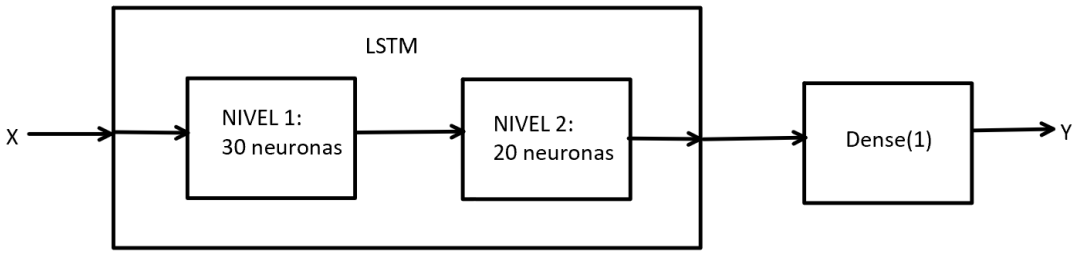


Figura 14. Modelo LSTM

En el diagrama anterior, X representa la entrada del modelo e Y la salida. La entrada del modelo pasa por dos niveles LSTM, el primero con más neuronas que el segundo. La salida de esta parte se pasa a la última caja del diagrama (Dense(1)), la cual representa la función de activación, que en este caso se trata de una capa Dense con un tamaño de salida de una dimensión, ya que queremos predecir el siguiente valor del índice *Kp*. El "1" que lleva la capa Dense entre paréntesis representa esa única dimensión de salida que queremos.

Para poder dar por terminado el modelo, faltaría elegir unos parámetros que optimicen sus predicciones. Estos parámetros son el número de épocas durante las que entrenar el modelo, el número de neuronas de cada capa LSTM, y el tamaño del lote (*batch size*). Para optimizar los parámetros, nos fijamos en las pérdidas del modelo en función de cada uno de los parámetros, además del error cuadrático medio (RMSE) y del error medio y la desviación típica de ese error. En concreto, tras entrenar un modelo con ciertos parámetros, observamos y comparamos con modelos anteriores las siguientes gráficas:

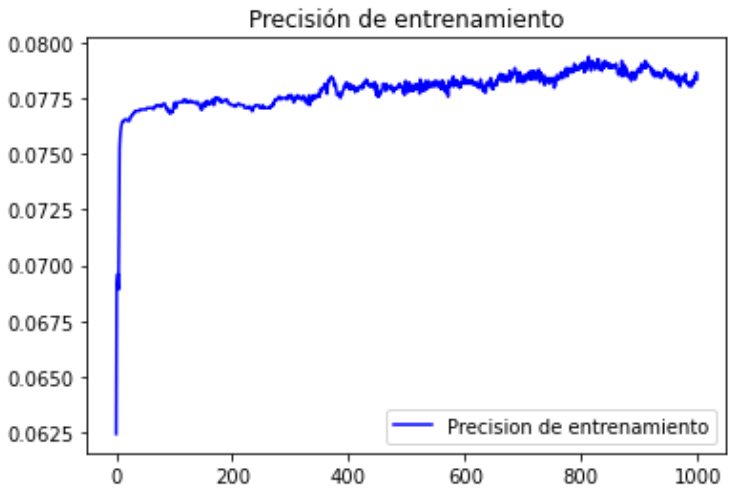


Figura 15. Precisión de entrenamiento en función del número de épocas LSTM

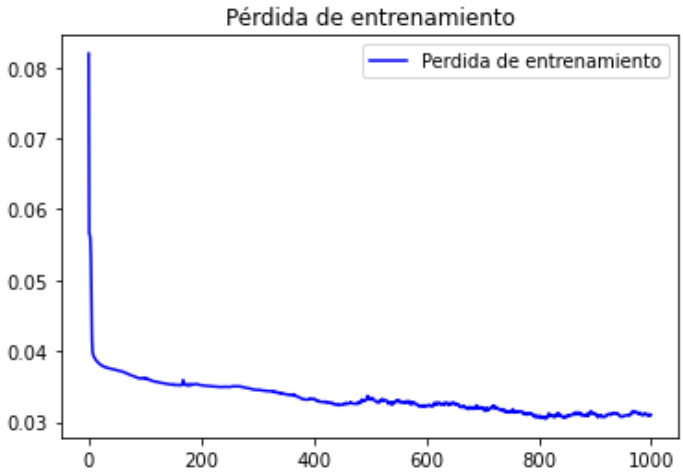


Figura 16. Pérdida de entrenamiento en función del número de épocas LSTM

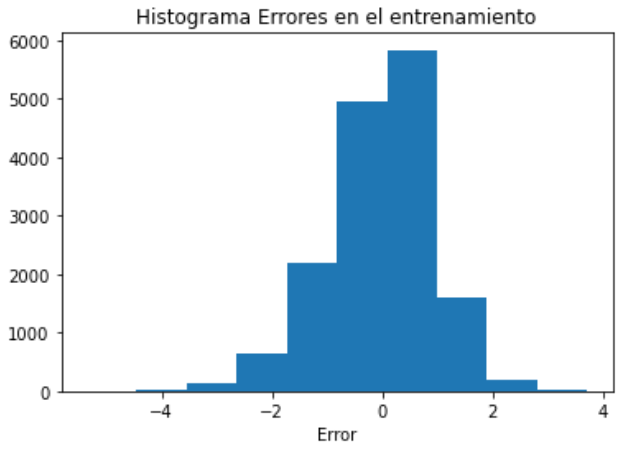


Figura 17. Histograma de errores en el entrenamiento LSTM

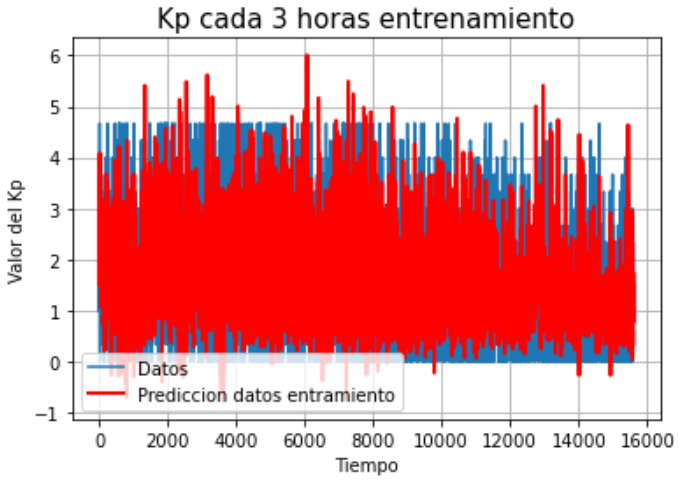


Figura 18. Predicciones del Kp en entrenamiento LSTM

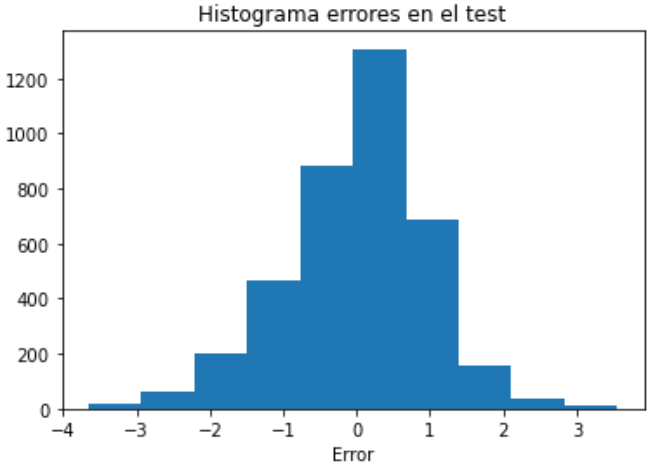


Figura 19. Histograma de errores en el test LSTM

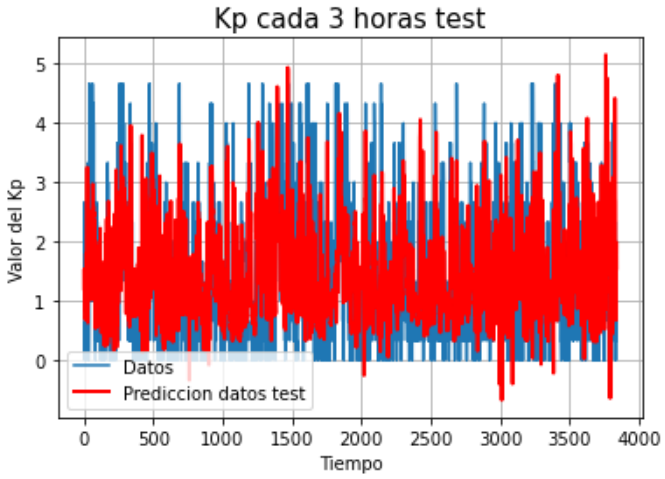


Figura 20. Predicciones del Kp en test LSTM

```
error medio de entrenamiento: -0.0299
desviación típica de entrenamiento: 0.9558
```

Figura 21. Error medio y desviación típica en entrenamiento LSTM

```
error medio de test: 0.0030
desviación típica de test: 0.9627
```

Figura 22. Error medio y desviación típica en el test LSTM

```
Resultado del entrenamiento: 0.9563 RMSE
Resultado del test: 0.9627 RMSE
```

Figura 23. RMSE de entrenamiento y de test LSTM

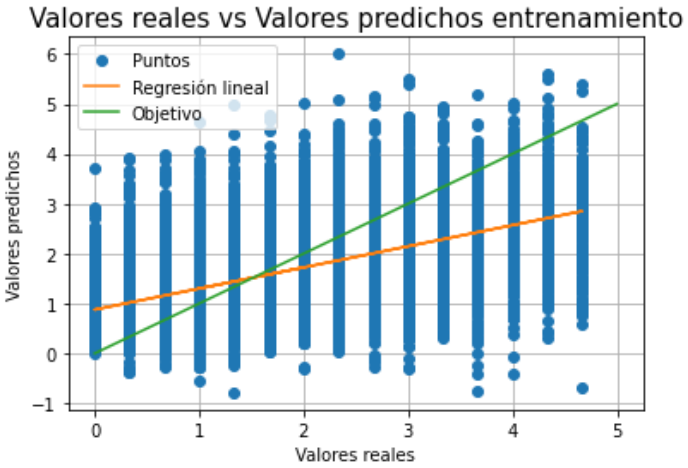


Figura 24. Valores reales vs valores predichos en entrenamiento LSTM

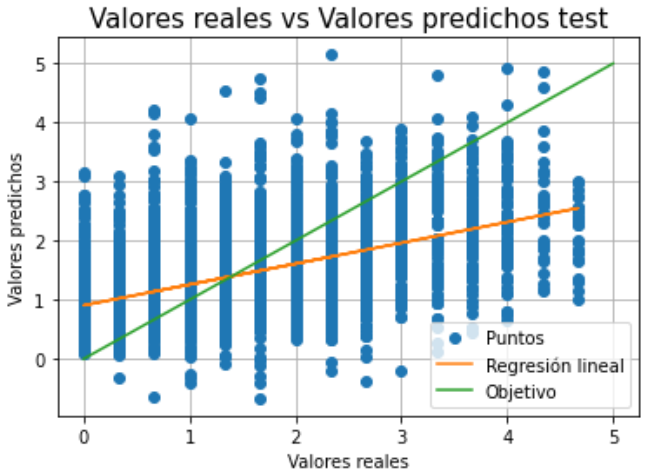
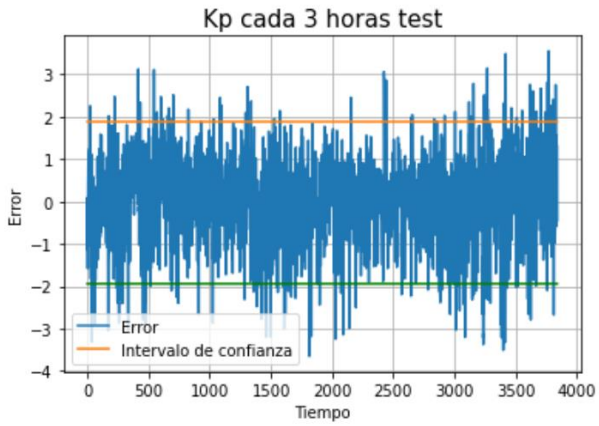


Figura 25. Valores reales vs valores predichos en test LSTM




```
94.78895257946847% de predicciones en el intervalo de confianza
```

Figura 26. Errores respecto al intervalo de confianza LSTM

Las figuras Figura 15 y Figura 16**Error! Reference source not found.** deben tener esa forma de L invertida y de L. Esto significa que mejora el modelo con mayor número de épocas. Sin embargo, se puede observar que a partir de 800 épocas en la Figura 16, la pérdida deja de decrecer y se mantiene estable. Con mil épocas, obtenemos un resultado que se sitúa en la zona óptima. Respecto a los histogramas de las figuras Figura 17 y Figura 19**Error! Reference source not found.**, en ellos queremos conseguir un error medio cercano a cero con poca varianza. Por otro lado, en las figuras Figura 18 y Figura 20 observamos directamente cómo de bien se ajustan las predicciones del modelo a los datos originales tanto en entrenamiento como en test. En las figuras Figura 24 y Figura 25 se comparan los valores reales y los valores predichos por el modelo. En las gráficas se muestra una recta de color naranja, que se consigue llevando a cabo una regresión lineal de los puntos. En estas figuras, el objetivo es que esta regresión lineal se aproxime lo máximo posible a la línea marcada en verde, que tiene una inclinación de 45 grados. Finalmente, la Figura 26 representa los errores de las predicciones y las bandas de confianza correspondientes. Estas bandas son calculadas sumándole y restándole al error medio del entrenamiento dos veces la desviación típica. El objetivo es contener la gran mayoría de los errores dentro de estas bandas.

Realizando varias pruebas con distintos valores de parámetros obtenemos los siguientes resultados:

Modelos	Capas	Neuronas	Tamaño de lote	Épocas	Error Medio Training	Desviación Típica Training	Error Medio Test	Desviación Típica Test	RMSE Training	RMSE Test
M1	2	30,20	200	500	-0.0361	0.7506	-0.0196	0.7458	0.7515	0.7461
M2	2	30,20	200	1000	-0.103	0.8182	-0.035	0.7874	0.8247	0.7882
M3	2	30,20	200	2000	-0.0069	0.8661	0.0068	0.8495	0.8661	0.8496
M4	2	15,5	200	500	0.0048	0.7138	0.0098	0.7263	0.7138	0.7263
M5	2	15,5	200	1000	0.0023	0.7136	-0.0001	0.725	0.7136	0.725
M6	2	15,5	200	2000	0.0069	0.7286	0.0137	0.7405	0.7286	0.7406
M7	2	20,10	200	500	0.0421	0.7292	0.0455	0.7379	0.7304	0.7393
M8	2	20,10	200	1000	0.0064	0.7573	0.0149	0.7591	0.7574	0.7592
M9	2	20,10	200	2000	0.0195	0.8291	0.0478	0.8596	0.8294	0.8609
M10	2	30,20	100	1000	0.0322	0.8248	0.0078	0.8035	0.8254	0.8035
M11	2	30,20	300	1000	0.0759	0.8821	0.1099	0.867	0.8853	0.874

Figura 27. Resultados para diferentes parámetros LSTM

Para el cálculo de estos resultados se ha incluido el propio índice *Kp* como una de las entradas al modelo. Observando la tabla, vemos que obtenemos los mejores resultados para los modelos M5 y M2, los cuales aparecen resaltados en la figura anterior (Figura 27). En base a los datos mostrados, se puede afirmar que las predicciones del M5 son más ajustadas, ya que tienen el menor error medio cuadrático (RMSE) de test de todos los modelos probados. Sin embargo, dado que el objetivo del modelo es que prediga precisamente valores del *Kp* menores de cinco y que se equivoque con valores mayores de cinco, debemos tener en cuenta el funcionamiento del modelo a la hora de predecir valores correspondientes a periodos de tormentas. Para ello, predecimos valores correspondientes a

periodos de tormenta solar, calculamos el error en la predicción y comprobamos si se encuentra dentro del intervalo de confianza de acierto del modelo. Los resultados del modelo M5 son los siguientes:

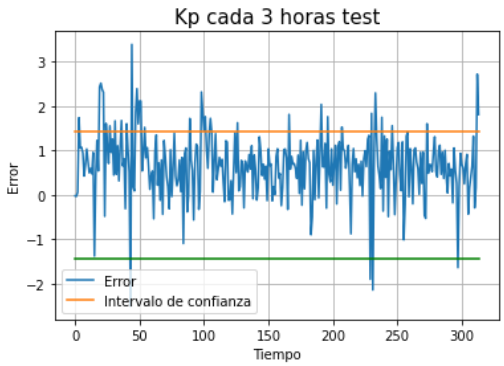


Figura 28. Errores en la predicción del M5 en periodo de tormentas

Tal y como se puede observar en la gráfica, la mayor parte de los errores en la predicción se sitúan dentro del intervalo de confianza del modelo. En concreto, el 87,58% de los errores no se detectan como anomalías. Teniendo en cuenta el objetivo del trabajo, este porcentaje es demasiado alto para detectar tormentas correctamente. Por ello, se decide no incluir el propio índice *Kp* como entrada del modelo, ya que se deduce que se le da demasiado peso a la hora de realizar las predicciones. De esta forma, comprobamos el funcionamiento de los dos mejores modelos (M2 y M5) sin incluir el *Kp* en las entradas del modelo. A continuación, se pueden observar los resultados:

Modelos	Capas	Neuronas	Tamaño de lote	Épocas	Error Medio Training	Desviación Típica Training	Error Medio Test	Desviación Típica Test	RMSE Training	RMSE Test
M5	2	15,5	200	1000	-0.0007	0.9007	0.0169	0.8795	0.9007	0.8797
M2	2	30,20	200	1000	-0.0299	0.9558	0.003	0.9627	0.9563	0.9627

Figura 29. Resultados M5 y M2 sin Kp en la entrada LSTM

Como era de esperar, ambos modelos pierden precisión a la hora de predecir valores de *Kp* menores de cinco. Sin embargo, a cambio conseguimos los siguientes resultados a la hora de predecir valores de *Kp* en momentos de tormenta:

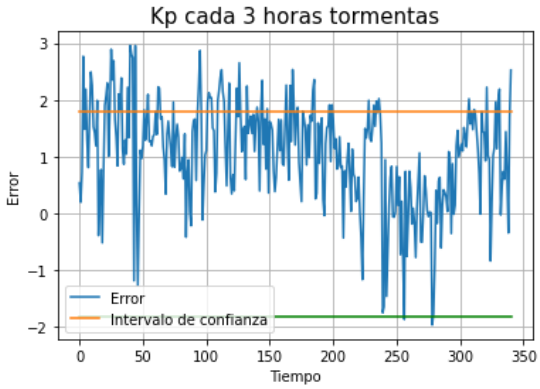


Figura 30. Errores en la predicción M5 sin Kp en la entrada

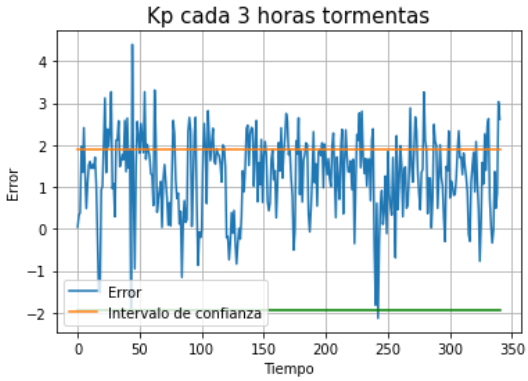


Figura 31. Errores en la predicción M2 sin Kp en la entrada

Se puede observar a simple vista que disminuye notablemente el número de errores que se encuentran en el intervalo de confianza del modelo para las predicciones de tormentas electromagnéticas. En concreto, el porcentaje del M5 desciende hasta un 80,05%% y el porcentaje del M2 desciende hasta un 71,85%. A la vista de estos porcentajes, elegimos los parámetros correspondientes al modelo M2: 1.000 épocas, con treinta neuronas en la primera capa LSTM, con veinte neuronas en la segunda capa LSTM y con un tamaño de lote de 200. Cabe destacar que el objetivo del modelo es predecir el valor del *Kp* del siguiente intervalo de tres horas. Sin embargo, con un tamaño de lote de 200, estaríamos prediciendo los 200 valores siguientes a partir de los 200 anteriores. Por este motivo, después de entrenar el modelo con un tamaño de lote de 200, guardamos los pesos del modelo y los cargamos en uno exactamente igual, pero con tamaño de lote de 1. Este nuevo modelo es el que se usará para realizar las predicciones.

4.2 Modelo convolucional

El proceso de creación de una red convolucional es parecido al descrito en el apartado anterior para una red LSTM. En primer lugar, importamos los datos a un *dataframe*. Después los normalizamos y los dividimos en conjuntos de entrenamiento y de test de la misma manera que en el apartado anterior.

El siguiente paso es la creación de la red neuronal, que consistirá en tres capas de convolución seguidas de una capa de activación lineal:

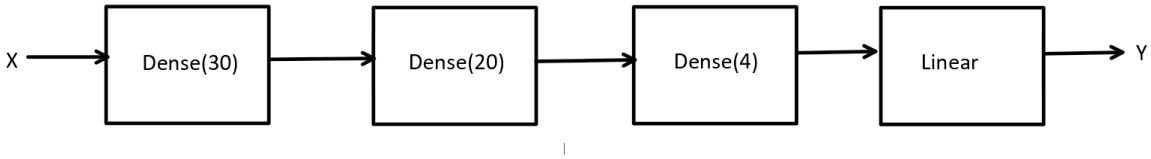


Figura 32. Red convolucional

En concreto, cada capa *Dense* tendrá un número de neuronas. Como anteriormente, X representa las entradas e Y la salida. Teniendo en cuenta lo aprendido con la red LSTM respecto al peso que se le da al *Kp* si se usa como entrada, este modelo tampoco lo incluirá en las entradas. Por tanto, las entradas serán, al igual que en el modelo LSTM, la magnitud del campo magnético interplanetario, la densidad de protones del viento solar, y su velocidad de flujo. Además, para este número de neuronas obtenemos el siguiente resumen de la red:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 30)                120

 dense_1 (Dense)             (None, 20)                620

 dense_2 (Dense)             (None, 4)                 84

 activation (Activation)     (None, 4)                 0

=================================================================
Total params: 824
Trainable params: 824
Non-trainable params: 0
_________________________________________________________________
```

Figura 33. Resumen del modelo convolucional

Finalmente, hay que optimizar los parámetros de la red, que en este caso son las épocas de entrenamiento, las neuronas de cada capa convolucional y el tamaño de lote. Para ello, estudiamos el efecto de los parámetros con las mismas gráficas que en el modelo LSTM:

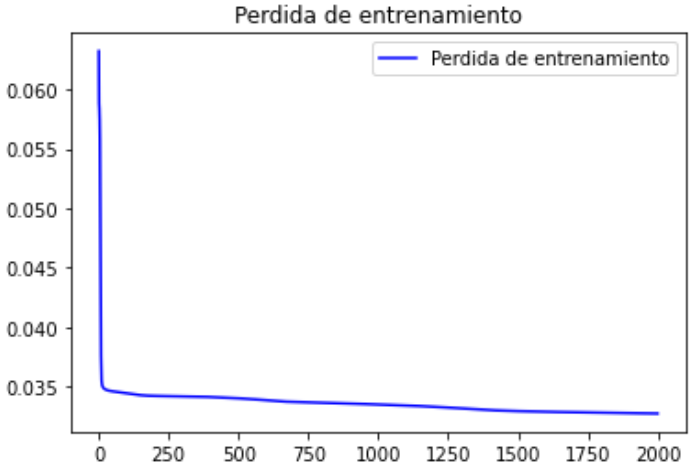


Figura 34. Pérdida de entrenamiento en función del número de épocas del modelo convolucional

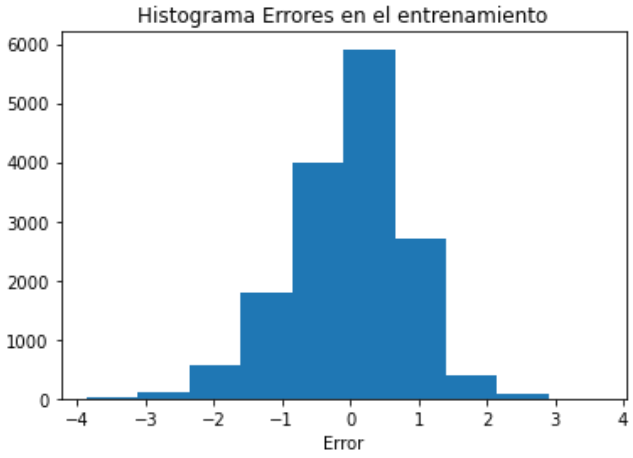


Figura 35. Histograma de errores en entrenamiento del modelo convolucional

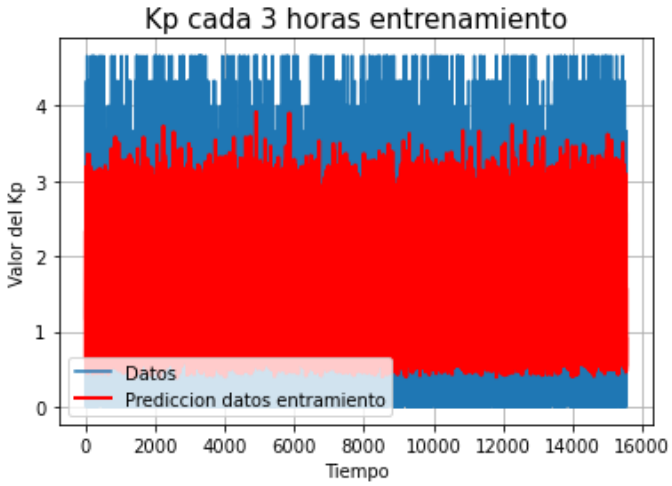


Figura 36. Predicciones del Kp en entrenamiento del modelo convolucional

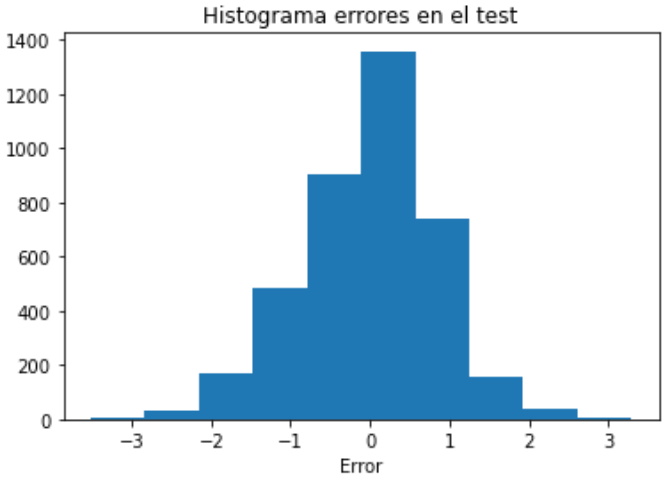


Figura 37. Histograma de errores en el test del modelo convolucional

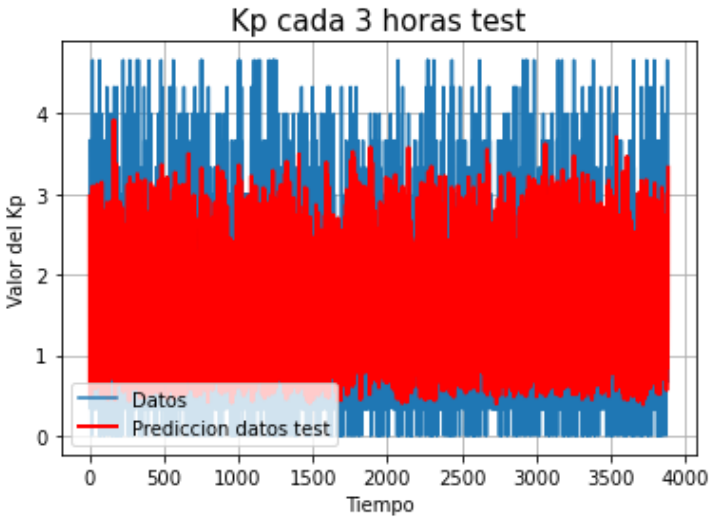


Figura 38. Predicciones del Kp en test del modelo convolucional

```
error medio de entrenamiento: -0.0025
desviación típica de entrenamiento: 0.8419
```

Figura 39. Error medio y desviación típica del entrenamiento del modelo convolucional

```
error medio de test: -0.0407
desviación típica de test: 0.8590
```

Figura 40. Error medio y desviación típica del test del modelo convolucional

```
Resultado del entrenamiento: 0.8419 RMSE
Resultado del test: 0.8600 RMSE
```

Figura 41. RMSE de entrenamiento y de test del modelo convolucional

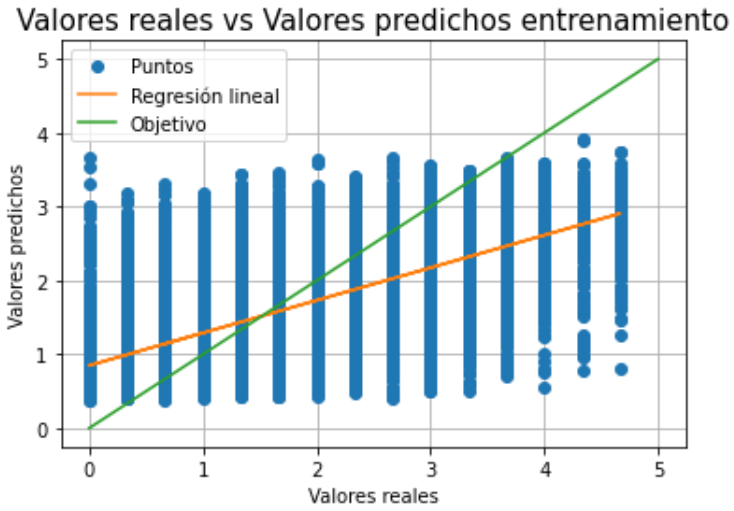


Figura 42. Valores reales vs valores predichos entrenamiento del modelo convolucional

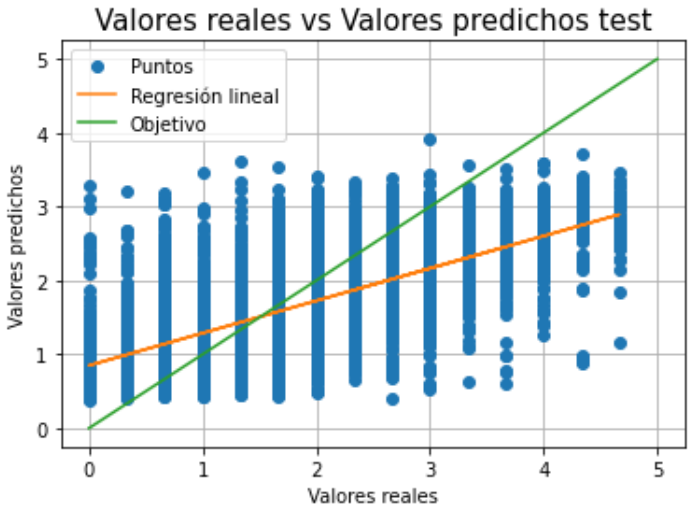
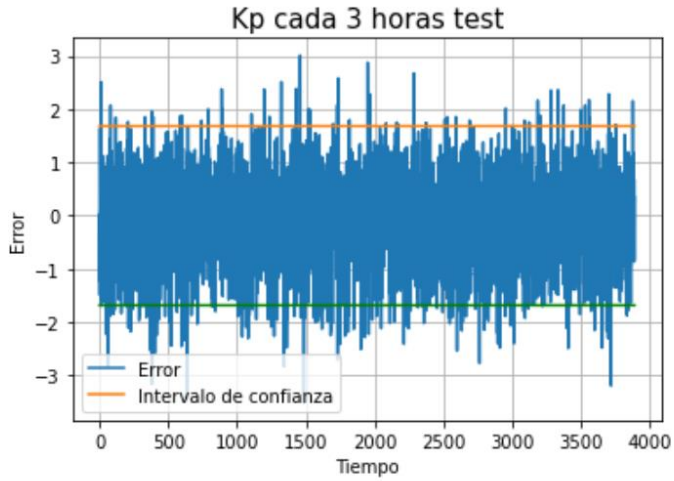


Figura 43. Valores reales vs valores predichos test del modelo convolucional




```
94.70300848547184% de predicciones en el intervalo de confianza
```

Figura 44. Errores respecto al intervalo de confianza del modelo convolucional

Tal y como con el modelo LSTM, la Figura 34 tiene una forma de "L" que se aplana al llegar a las 2.000 épocas. Por otro lado, se pueden observar distribuciones normales en los histogramas de errores (Figura 35 y Figura 37), tanto de entrenamiento como de test. Este es el comportamiento esperado, en ambos casos con un error medio cercano a cero y una desviación típica relativamente pequeña. Además, en las figuras Figura 42 y Figura 43 se puede apreciar que la regresión lineal de los puntos se aproxima suficientemente a la línea objetivo que se muestra en verde. De todas formas, se comprueban los resultados del modelo para distintos parámetros:

Modelos	Capas	Neuronas	Tamaño de lote	Épocas	Error Medio Training	Desviación Típica Training	Error Medio Test	Desviación Típica Test	RMSE Training	RMSE Test
M12	3	30,20,4	200	1000	-0.0272	0.8855	-0.012	0.8936	0.8859	0.8937
M13	3	30,20,4	200	500	-0.0239	0.8895	-0.0052	0.8963	0.8898	0.8963
M14	3	30,20,4	200	2000	-0.0239	0.8743	-0.0115	0.887	0.8746	0.887
M15	3	30,20,4	100	1000	-0.0135	0.8807	0.0006	0.8885	0.8808	0.8885
M16	3	30,20,4	100	2000	-0.0025	0.8419	-0.0407	0.859	0.8419	0.86
M17	3	30,20,4	100	4000	-0.0022	0.8627	0.0045	0.888	0.8627	0.888
M18	3	15,5,4	100	2000	-0.0071	0.888	0.0105	0.8966	0.888	0.8966

Figura 45. Resultados para diferentes parámetros del modelo convolucional

Cómo era de esperar según las gráficas comentadas en el párrafo anterior, obtenemos las predicciones más precisas para el modelo M16, que consta de 2.000 épocas de entrenamiento, treinta neuronas en la primera capa *Dense*, veinte neuronas en la segunda capa *Dense*, cuatro neuronas en la tercera capa *Dense* y con un tamaño de lote de 100.

4.3 Detección de anomalías

En este apartado se expone como se hace uso de los modelos descritos en los puntos 4.1 y 4.2 para la detección de tormentas. En primer lugar, debemos utilizar los modelos optimizados para hacer predicciones de datos recogidos en periodos de tormentas. La idea es que, como los modelos están entrenados en periodos en los que no tienen lugar tormentas electromagnéticas, deberían predecir erróneamente en periodos en los que sí haya tormentas solares. Para comprobar en qué cantidad esto es así, pasaremos por los modelos datos correspondientes a periodos en los que hay tormentas y observaremos los resultados.

4.3.1 Modelo LSTM en periodo de tormentas

En primer lugar, hablaremos sobre el modelo LSTM. De manera similar a cuando entrenamos el modelo, lo primero es descargar los datos correspondientes a periodos de tormentas solares y copiarlos en un *dataframe* de la librería *pandas* [10]. Después, se normalizan los datos haciendo uso del *MinMaxScaler* de la librería *sklearn* [17]. Una vez tenemos los datos normalizados, los utilizamos como datos de test y hacemos una predicción sobre ellos con el modelo LSTM. Finalmente, comparamos los errores de las predicciones respecto a los datos originales, y los superponemos a las bandas de confianza que usamos anteriormente en la Figura 26. El resultado es el siguiente:

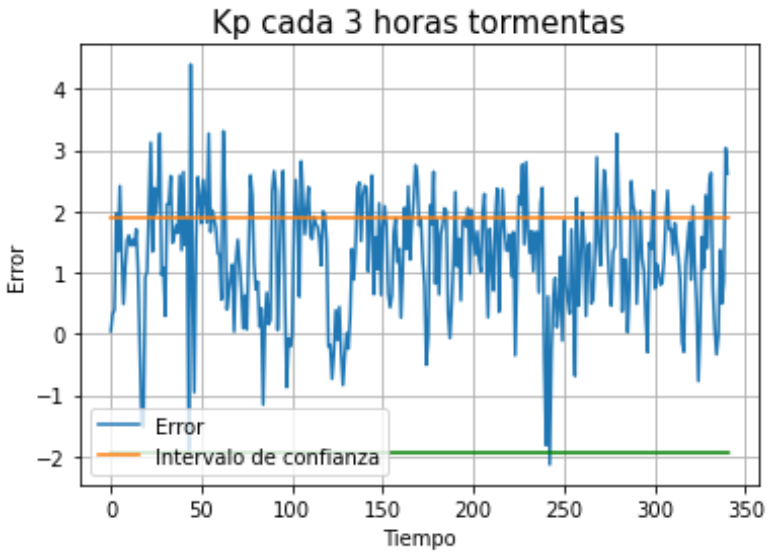


Figura 46. Errores respecto al intervalo de confianza en tormentas LSTM

Como era de esperar, hay un mayor porcentaje de errores fuera del intervalo de confianza que en el caso de los datos correspondientes a periodos de no tormentas. En concreto, en este caso el 71,85% de los errores se encuentran dentro de ese intervalo de confianza, frente al 94,79% que lo hacía con los datos de test.

4.3.2 Modelo convolucional en periodo de tormentas

Para comprobar los resultados del modelo convolucional en periodos de tormentas solares, llevamos a cabo un procedimiento igual que para el modelo LSTM. En primer lugar, copiamos los datos con las variables de entrada y la variable de salida. Segundo, normalizamos esos datos y los utilizamos como datos de test. Tercero, hacemos una predicción con el modelo optimizado elegido en el apartado 4.2. Finalmente, calculamos el error de cada predicción respecto a su valor objetivo y comprobamos si se sitúa dentro el intervalo de confianza de error del modelo.

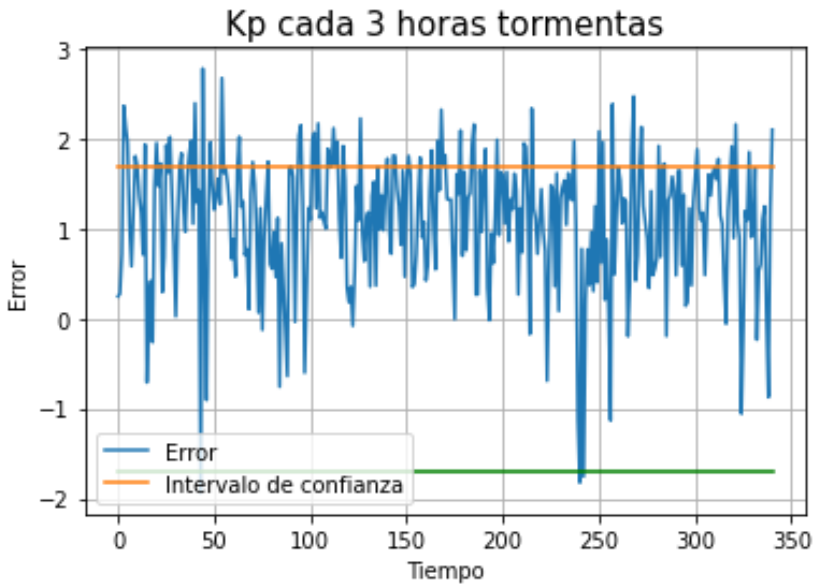


Figura 47. Errores respecto al intervalo de confianza en tormentas modelo convolucional

En la figura se observa, al igual que con el modelo LSTM, que un mayor número de predicciones se alejan del valor real más allá del intervalo de confianza, comparado con los datos de test. En concreto, en este caso un 78,29% de los errores se encuentra dentro de las bandas de confianza. Por el contrario, el 94,7% de los datos medidos en periodos de no tormentas se encuentra dentro del intervalo de confianza, tal y como se puede ver en la Figura 26.

4.4 Elección del modelo

En este apartado se elige un modelo para ser utilizado en la detección de tormentas y se explica las razones por las que es elegido. Basándonos en las figuras Figura 46 y Figura 47, el modelo LSTM es el que contiene un menor número de errores dentro del intervalo de confianza. Es decir, el modelo LSTM predice peor el valor próximo del *Kp*, cuando este corresponde a un periodo de tormenta. Sin embargo, el porcentaje de errores que se mantienen dentro de las bandas de confianza es demasiado alto tanto para la red LSTM como para el modelo convolucional. Por ello, se plantea una alternativa con la que obtenemos mejores resultados. En vez de utilizar un modelo u otro, se va a realizar un OR lógico de la salida de ambos modelos. Dado que el objetivo es calificar como momento con tormenta aquél para el que su predicción se aleja más del intervalo de confianza del objetivo, bastará que la predicción de cualquiera de los modelos se salga de sus correspondientes bandas de confianza para marcar el suceso como una anomalía. El resultado es el siguiente:

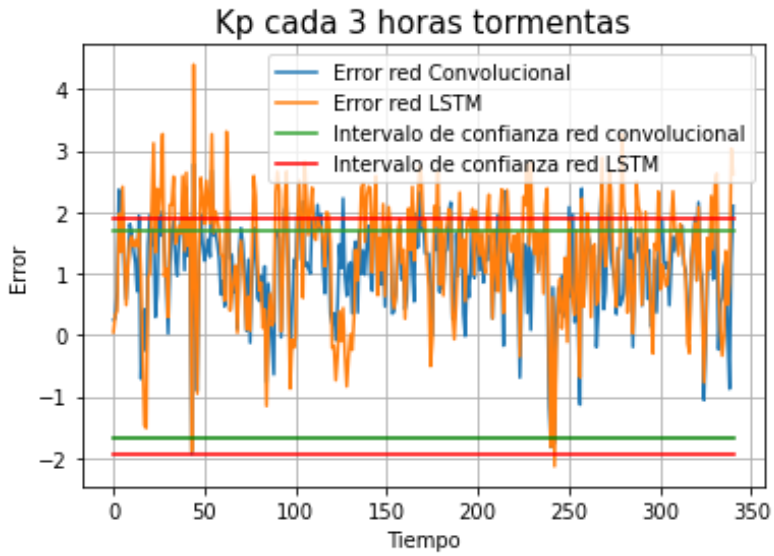


Figura 48. Errores respecto al intervalo de confianza en tormentas OR de los modelos

Se puede observar que aumenta el número de errores fuera del intervalo, en concreto, el número de errores en la predicción que se mantiene dentro de las bandas de confianza se reduce al 62,75%. Por lo tanto, conseguimos una mejoría considerable. Reducimos ese porcentaje más de un 15% respecto al que conseguíamos con el modelo convolucional, y casi un 10% respecto al que conseguíamos con el modelo LSTM. Sin embargo, antes de elegir este "modelo OR" para la detección de anomalías, debemos comprobar que no se vea perjudicado el porcentaje de errores en la predicción de datos en periodos en los que no hay tormentas. Para ello, realizamos el OR también con los datos de no tormentas y obtenemos la siguiente gráfica:

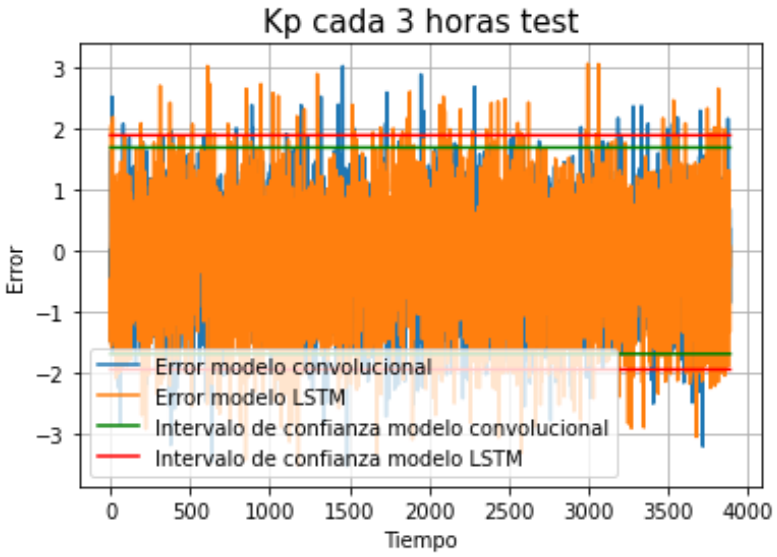


Figura 49. Errores respecto al intervalo de confianza OR de los modelos

A simple vista no parece haber una disminución notable de los errores que se quedan fuera del intervalo de confianza. Concretamente, el 93,21% de los errores en las predicciones se mantienen dentro de las bandas de confianza. Tan solo disminuye un 1% con respecto a los resultados de los modelos por separado. Con estos resultados, podemos afirmar que el "modelo OR" resulta más efectivo que el LSTM o el convolucional por su cuenta.

4.5 Aplicación para la detección de tormentas

En este apartado se expone la creación y el funcionamiento de la aplicación "semáforo", encargada de detectar si ha habido tormenta o no en un día o mes seleccionado. La aplicación hace uso de la librería *ipywidgets* [11], la cual facilita el uso de controles interactivos. En concreto, la aplicación dispone de dos menús desplegables para la elección de la fecha o fechas para las que queramos ver si ha habido tormentas electromagnéticas. El primero de ellos contiene todos los meses de todos los años desde 2015 hasta 2021 y tiene este aspecto:

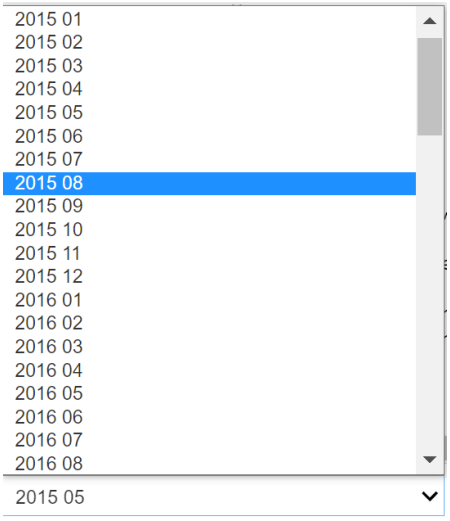


Figura 50. Menú desplegable de meses

Por otro lado, el segundo menú desplegable contiene los días del mes elegido, y una opción que permite visualizar todos los días del mes. El menú tiene el siguiente aspecto:

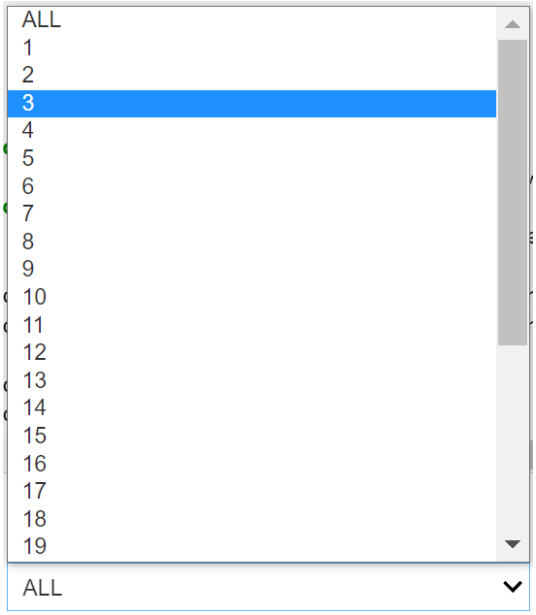


Figura 51. Menú desplegable con los días del mes

Respecto a la computación realizada por la aplicación, lo primero que se lleva a cabo es la predicción de todas las fechas del calendario con ambos modelos. De esta manera, solo hay que realizar las predicciones una sola vez y una vez hechas la aplicación puede funcionar sin pausas de computación para realizar las predicciones que es lo que más tiempo consume. Una vez hechas las predicciones, se calculan los errores de estas. Después, se utilizan las selecciones de los menús desplegables para filtrar todos los datos de manera que se elijan solamente las fechas deseadas. Una vez sabemos para que días queremos saber si ha habido tormenta solar, pasamos los errores en las predicciones de esos días por el detector de anomalías. Es decir, comprobamos si esos errores se encuentran dentro del intervalo de confianza. Como finalmente elegimos utilizar el "modelo OR", tendremos que comprobar los errores en la predicción de cada modelo comparados con sus respectivas bandas de confianza. Una vez hecho esto, mostramos los resultados gráficamente en función del número de anomalías que haya en un día.

Los resultados siguen un código de colores para indicar la certeza con la que afirmamos que ha habido tormenta o no ese día. Estos colores son verde, amarillo, rojo y gris. Inspirados en los colores de un semáforo, asignamos al verde los días en los que la aplicación calcula que no hay tormentas. Por otro lado, el amarillo representa días con mayor probabilidad de haber sufrido una tormenta, y el rojo los días en los que afirmamos con mayor certeza que ha habido una tormenta. Por último, asignamos el gris a los días en los que alguna entrada fue desechada en el proceso de tratamiento de datos.

El planteamiento inicial es que, si hay más de cinco anomalías, marcar el día de rojo, si hay entre tres y cinco, marcarlo de amarillo, si hay menos de tres marcarlo de verde y si es un día en el que faltan

entradas, marcarlo de gris. Sin embargo, el resultado para esta configuración no es satisfactorio. Para respaldar esta afirmación, podemos ver los resultados del modelo para el mes de marzo de 2015:

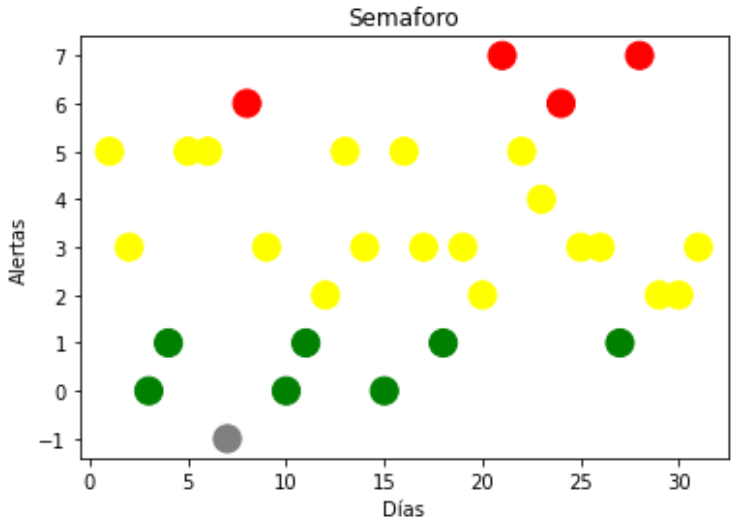


Figura 52. Versión inicial del semáforo en marzo de 2015

Como se puede observar, obtenemos días de todos los colores por lo que es un buen ejemplo en el que fijarnos. Tomemos por ejemplo el día 17 de marzo, que sale coloreado de amarillo por la aplicación, con tres anomalías. Estos son los datos correspondientes al 17 de marzo:

	YearMonth	Day	Hours	Kp	Bt	Proton Density	Bulk Speed
593	2015 03	17	0:3	2.000	8.300000	14.100000	420.266667
594	2015 03	17	3:6	4.667	15.033333	17.433333	468.966667
595	2015 03	17	6:9	5.667	14.166667	6.266667	530.466667
596	2015 03	17	9:12	5.333	15.766667	7.000000	609.566667
597	2015 03	17	12:15	7.667	27.133333	13.700000	589.866667
598	2015 03	17	15:18	7.667	24.400000	10.066667	576.033333
599	2015 03	17	18:21	7.333	17.400000	5.033333	556.833333
600	2015 03	17	21:24	7.667	19.166667	5.766667	555.200000

Figura 53. Datos del 17 de marzo de 2015

Se puede apreciar en la tabla, que el 17 de marzo de 2015 fue un día con muchas tormentas. Teniendo en cuenta que el índice *Kp* solamente va de cero a nueve, obtener *Kp*s de magnitud siete es algo bastante notable. Sin embargo, la aplicación solamente detecta tres anomalías en un día en el que seis de los ocho intervalos de tres horas corresponden a periodos de tiempo en los que ocurren tormentas solares. Este día debería ser calificado de rojo. Por otro lado, podemos fijarnos en un caso contrario. El

día 24 de marzo, la aplicación lo marca como rojo, con seis anomalías. Estos son los valores correspondientes al 24 de maro de 2015:

	YearMonth	Day	Hours	Kp	Bt	Proton Density	Bulk Speed
649	2015 03	24	0:3	1.667	5.700000	1.933333	518.133333
650	2015 03	24	3:6	1.333	5.200000	2.533333	517.133333
651	2015 03	24	6:9	1.333	4.200000	2.700000	540.566667
652	2015 03	24	9:12	2.333	3.600000	2.400000	536.466667
653	2015 03	24	12:15	3.333	3.866667	3.100000	530.900000
654	2015 03	24	15:18	3.667	4.600000	2.933333	527.533333
655	2015 03	24	18:21	1.667	5.933333	2.500000	524.866667
656	2015 03	24	21:24	2.000	4.200000	2.766667	551.066667

Figura 54. Datos del 24 de marzo de 2015

Como se observa en los datos mostrados, el 24 de marzo no tiene ningún intervalo de tres horas en el que haya tormenta solar, de hecho, ningún valor del *Kp* registrado ese día se acerca a valores correspondientes a periodos de tormentas. Lo mismo ocurre el 8, 21 y 28 del mismo mes. Estos días se marcan en rojo a pesar de no superar un *Kp* de 4 en ningún momento del día.

Vistos los resultados, planteamos otra opción para calcular las anomalías. Tanto el modelo LSTM como el modelo convolucional están entrenados para predecir valores de *Kp* menores de cinco, es decir, están entrenados en periodos en los que no hay tormentas. Por ello, a la hora de predecir un valor de tormenta, la predicción de los modelos será, generalmente, menor que el valor original. El error en la predicción se calcula restando al valor predicho el valor real. En consecuencia, en caso de salirse ese error de las bandas de confianza, normalmente lo hará por la banda inferior ya que la diferencia entre el valor predicho y el valor real será negativa. Por tanto, ajustamos la detección de anomalías para que detecte anomalías solamente por la banda inferior del intervalo de confianza. A raíz de este cambio, hay que ajustar también la asignación de colores. Para esta versión, se marcará de rojo aquel día en el que se detecten dos o más anomalías, de amarillo aquel en el que se detecte una, de verde aquel día en el que no se detecte ninguna y de gris aquel día en el que falten entradas. El resultado que obtenemos para el mismo mes aplicando estas modificaciones es el siguiente:

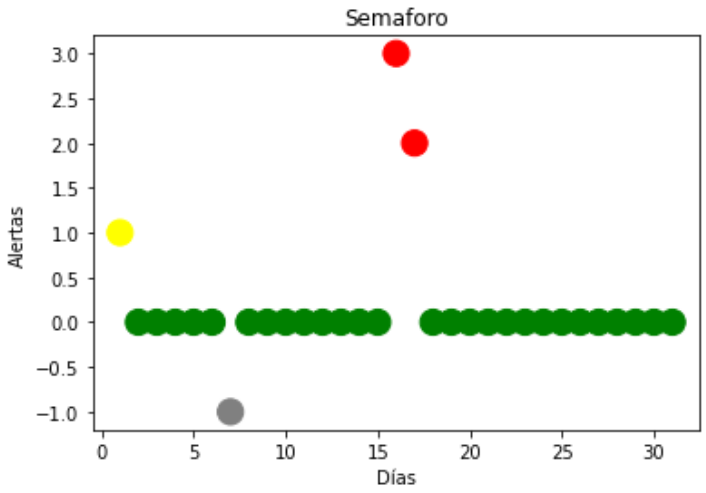


Figura 55. Versión final del semáforo en marzo de 2015

Como se puede observar, obtenemos un día amarillo, dos rojos, uno gris y el resto verdes. Si volvemos a la Figura 53, podemos observar que la aplicación califica correctamente el día 17 de marzo. Por otro lado, los días 8, 21, 24 y 28, los cuales eran clasificados erróneamente, pasan a ser clasificados como verdes, por lo que se clasifican correctamente. Por otro lado, el primer día del mes se sigue clasificando de amarillo. Podemos observar los valores del *Kp* ese día:

	YearMonth	Day	Hours	Kp	Bt	Proton Density	Bulk Speed
468	2015 03	1	0:3	5.000	10.266667	6.066667	472.300000
469	2015 03	1	3:6	5.000	8.633333	6.200000	520.633333
470	2015 03	1	6:9	5.333	6.400000	4.866667	535.066667
471	2015 03	1	9:12	3.667	6.366667	4.166667	547.633333
472	2015 03	1	12:15	2.000	9.233333	4.733333	521.500000
473	2015 03	1	15:18	1.333	10.000000	5.233333	504.700000
474	2015 03	1	18:21	2.333	7.200000	5.166667	496.133333
475	2015 03	1	21:24	2.667	6.533333	7.066667	454.800000

Figura 56. Datos del 1 de marzo de 2015

Se puede observar que hay dos valores de *Kp* iguales a cinco, y uno igual a 5.333. Estos valores corresponden a tormentas, aunque se sitúan en la frontera entre tormenta y no tormenta. Por ello, dado que el resto de los valores del día son *Kp*s bajos, parece que el amarillo es un color adecuado para este día. Otro día destacable de los nuevos resultados del semáforo, es el día 16. Este día se clasificaba de amarillo en la versión anterior y se clasifica de rojo con esta. Sin embargo, el día 16 contiene los siguientes datos:

	YearMonth	Day	Hours	Kp	Bt	Proton Density	Bulk Speed
585	2015 03	16	0:3	1.667	4.700000	19.900000	338.333333
586	2015 03	16	3:6	3.333	6.000000	20.166667	356.666667
587	2015 03	16	6:9	3.667	9.700000	12.966667	365.966667
588	2015 03	16	9:12	3.000	10.400000	19.166667	377.466667
589	2015 03	16	12:15	2.667	11.100000	11.300000	395.066667
590	2015 03	16	15:18	2.667	12.450000	14.050000	411.150000
591	2015 03	16	18:21	1.333	9.666667	18.366667	432.833333
592	2015 03	16	21:24	0.333	8.966667	13.733333	422.133333

Figura 57. Datos del 16 de marzo de 2015

Como se ve en la figura, ninguna entrada supera un *Kp* de cuatro. Por lo que este día se clasifica erróneamente. Sin embargo, teniendo en consideración la mejoría del resto de días, la aplicación afina considerablemente su precisión.

5. Análisis de Resultados

En este apartado, se analizan los resultados del trabajo y el cumplimiento de los objetivos marcados en el capítulo 1.3.

El primero objetivo, consistía en explicar la variable *Kp* en función de otras disponibles para elegir las entradas. En el apartado 3.1 se seleccionan las entradas que guardan una mayor correlación con el índice *Kp*, que son la magnitud total del campo magnético interplanetario, la densidad de protones en el viento solar, y la velocidad de su flujo. También se considera incluir el propio *Kp*, pero en el apartado 4.1, en la creación del modelo LSTM, se estima que se consigue un mejor comportamiento a la hora de detectar anomalías en las predicciones si se excluía de las entradas el *Kp*.

El segundo objetivo, era importar los datos de las variables elegidas, correspondientes a fechas desde el año 2015 hasta 2021. En el apartado 3.2, se expone detalladamente cómo se consiguen los datos de cada una de las variables de entrada.

El tercer objetivo, consistía en limpiar y tratar los datos recogidos. En el apartado 3.3, se detalla el método seguido para la interpolación de datos que no se habían medido correctamente. Además, después de interpolar o desechar las entradas de datos mal medidos, se unificaron los datos de todas las variables en tres archivos diferentes. Uno con todos los datos, otro con los datos de periodos de tormentas, y el último con datos de periodos en los que no había tormentas.

El cuarto objetivo se trataba de crear y evaluar la precisión de un modelo LSTM que prediga valores futuros del índice *Kp*. En el capítulo 2.1 y 2.1.1 se explica el funcionamiento de las redes neuronales recurrentes LSTM. Después, en el apartado 4.1, se detalla la creación del modelo y los pasos seguidos para su entrenamiento y optimización. Dentro de este apartado se muestran además todas las gráficas que nos dan la información necesaria acerca del buen funcionamiento del modelo para los parámetros elegidos. También se expone la arquitectura del modelo, explicando todas sus capas y parámetros. Finalmente, se llega a un modelo satisfactorio para las necesidades del proyecto y se comprueba su funcionamiento a la hora de detectar tormentas en el apartado 4.3.1.

El quinto objetivo, era análogo al anterior, pero para un modelo convolucional. En primer lugar, se explica el funcionamiento de las redes convolucionales en el apartado 2.2. Después, en el apartado 4.2, se detalla la creación del modelo y los pasos seguidos para su entrenamiento y optimización. Dentro de ese capítulo, se muestran también todas las gráficas utilizadas para determinar el buen funcionamiento del modelo para los parámetros elegidos. También se describe

la arquitectura del modelo, explicando todas sus capas y parámetros. Finalmente, se consigue un modelo satisfactorio para las necesidades del proyecto y se comprueba su funcionamiento a la hora de detectar tormentas en el apartado 4.3.2.

El sexto objetivo, se trataba de comparar el modelo LSTM con el modelo convolucional y determinar qué modelo era mejor a la hora de detectar tormentas. En el apartado 4.3, se discute la precisión con la que cada uno de los modelos detecta anomalías en sus predicciones en periodos de tormentas. Finalmente, se decide utilizar un modelo que haga un OR lógico de la detección de anomalías de ambos modelos para conseguir una mejor detección de anomalías. En el apartado 4.4 se detalla concretamente la mejoría en la detección de tormentas respecto a utilizar los modelos por separado, que es de aproximadamente un 10%.

El último objetivo, consistía en programar una aplicación que detecte tormentas electromagnéticas en la fecha seleccionada desde 2015 hasta 2021, basándose en los modelos de predicción desarrollados. En el apartado 4.5, se explica el proceso de desarrollo de la aplicación, así como su funcionamiento. Respecto al funcionamiento de la aplicación, del total de los datos usados por los modelos (19783), aproximadamente un 1.72% representa periodos en los que hay tormentas (341). Concretamente, el 98.2763% de los errores en la predicción del modelo OR, debería situarse dentro de su intervalo de confianza.

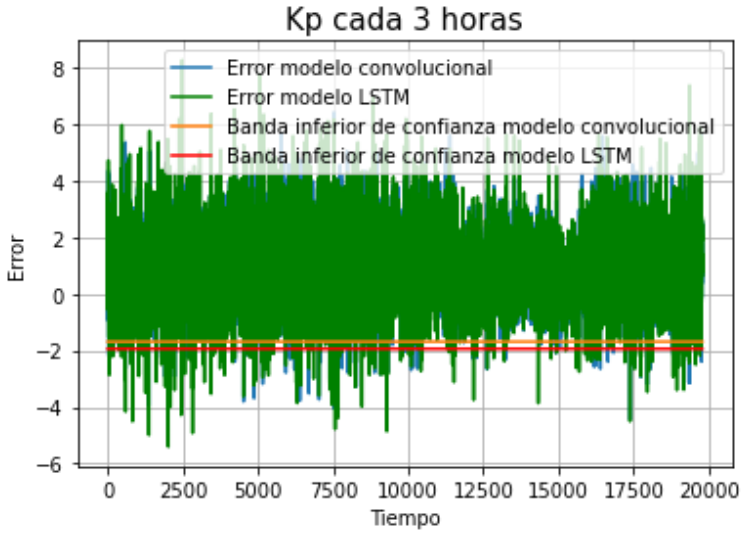


Figura 58. Errores en las predicciones de ambos modelos

En la gráfica superior se muestran los errores en las predicciones de todos los datos de ambos modelos. También se muestran las bandas inferiores de confianza de los modelos. Según estos datos, el 98.2864% de los errores se encuentran por encima de esas bandas. A pesar de que algunas

de estas anomalías no sean causadas por periodos de tormentas, clasificamos prácticamente el porcentaje de los datos esperados como anomalía o no anomalía.

Concretamente, de las cien anomalías que detecta la aplicación, 31 corresponden a periodos de tormentas. Sin embargo, hay que tener en cuenta que para que la aplicación marque un día de rojo, es decir, que afirme con cierta certeza que ha habido tormenta, debe de haber más de una anomalía ese día. Del total de los días, 2037 se marcan como verdes y en ninguno de ellos hay un valor de *Kp* superior o igual a cinco, por lo que se clasifican correctamente. Por lo tanto, en todos los casos en los que hay algún periodo de tormenta, el día es clasificado por la aplicación cómo amarillo o rojo. Por lo tanto, el resultado de la aplicación es satisfactorio y cumple con el objetivo marcado.

6. Conclusión

Tras todo el análisis realizado anteriormente, se puede afirmar que aún existe un cierto margen de mejora de la aplicación a la hora de detectar anomalías en las predicciones. Sin embargo, obtenemos un buen resultado con el modelo actual ya que conseguimos clasificar todos los momentos de tormentas como amarillos o rojos. La conclusión general es que se han abordado con éxito los objetivos del proyecto, citados en el apartado 1.3, obteniendo resultados satisfactorios en todos ellos. Se ha importado un conjunto de datos grande de varias variables que se han estudiado para ser incluidas como entradas de los modelos. También, se ha creado y optimizado un modelo LSTM y un modelo convolucional, en ambos casos teniendo en cuenta varios datos y gráficas y con numerosas pruebas a la hora de elegir los valores de los parámetros que optimizaban los modelos. Estos modelos se combinaron para obtener el mejor detector de anomalías posible y finalmente, se programó una aplicación interactiva en la que comprobar las anomalías en cualquier fecha desead

7. Bibliografía

[1] *Index of /sdb/goes/ace/monthly*. (2010). SPACE WEATHER PREDICTION CENTER. https://sohoftp.nascom.nasa.gov/sdb/goes/ace/monthly/

[2] *convenient ASCII format for Kp, ap, Ap, SN, F10.7 via FTP server*. (2021). GFZ German Research Centre for Geosciences. ftp://ftp.gfz-potsdam.de/pub/home/obs/Kp_ap_Ap_SN_F107

[3] T. Liu, T. Wu, M. Wang, M. Fu, J. Kang and H. Zhang, "Recurrent Neural Networks based on LSTM for Predicting Geomagnetic Field," 2018 IEEE International Conference on Aerospace Electronics and Remote

[4] Tan, Y., Hu, Q., Wang, Z., & Zhong, Q. (2018). Geomagnetic index Kp forecasting with LSTM. Space Weather, 16, 406–416

[5] Shibaji Chakraborty and Steven Karl Morley. Probabilistic prediction of geomagnetic storms and the Kp index. J. Space Weather Space Clim., 10 (2020) 36 DOI: https://doi.org/10.1051/swsc/2020037

[6] https://jupyter-notebook.readthedocs.io/en/v6.4.5/notebook.html

[7] https://www.python.org/downloads/release/python-397/

[8] keras RNN API https://keras.io/api/layers/recurrent_layers/lstm/

[9] https://matplotlib.org/3.4.3/contents.html

[10] https://pandas.pydata.org/pandas-docs/version/1.3.4/

[11] https://ipywidgets.readthedocs.io/en/7.6.5/

[12] https://numpy.org/devdocs/release/1.20.3-notes.html

[13] *2D Convolution*. (2018). [Figura]. https://arxiv.org/abs/1603.07285

[14] Matzka, J., Stolle, C., Yamazaki, Y., Bronkalla, O. and Morschhauser, A., 2021. The geomagnetic Kp index and derived indices of geomagnetic activity. Space Weather, https://doi.org/10.1029/2020SW002641

[15] U.S. Dept. of Commerce, NOAA, Space Weather Prediction Center. (2015, 31 enero). *Hourly Averaged Real-time Interplanetary Magnetic Field Values* [Conjunto de datos]. U.S. Dept. of Commerce, NOAA, Space Weather Prediction Center. https://sohoftp.nascom.nasa.gov/sdb/goes/ace/monthly/200008_ace_mag_1h.txt

[16] U.S. Dept. of Commerce, NOAA, Space Weather Prediction Center. (2015a, enero 31). *Hourly Averaged Real-time Bulk Parameters of the Solar Wind Plasma* [Conjunto de datos]. U.S. Dept. of Commerce, NOAA, Space Weather Prediction Center. https://sohoftp.nascom.nasa.gov/sdb/goes/ace/monthly/200008_ace_swepam_1h.txt

[17] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[18] https://scikit-learn.org/stable/whats_new/v0.24.html

[19] Torres, J. (2021, 22 marzo). *Redes Neuronales Recurrentes*. Jordi TORRES.AI. https://torres.ai/redes-neuronales-recurrentes/#:%7E:text=Las%20redes%20neuronales%20recurrentes%2C%20o,neuronales%20vistas%20en%20cap%C3%ADtulos%20anteriores.

[20] Juan Barrios, "Redes neuronales convolucionales" https://www.juanbarrios.com/redes-neurales-convolucionales/

[21] Ernest Scott Sexton, Katariina Nykyri and Xuanye Ma. *Kp forecasting with a recurrent neural network* J. Space Weather Space Clim., 9 (2019) A19 DOI: https://doi.org/10.1051/swsc/2019020

[22] Boberg, F., Wintoft, P., & Lundstedt, H. (2000). *Real time Kp predictions from solar wind data using neural networks, Physics and Chemistry of the Earth*. Science Direct. https://www.sciencedirect.com/science/article/abs/pii/S1464191700000167

[23] Objetivos de desarrollo sostenible. Recuperado el 9 de julio de 2021 de https://www1.undp.org/content/undp/es/home/sustainable-development-goals.htm

[24] Gamez, M. (2021). Objetivos y metas de desarrollo sostenible. Recuperado 9 de julio 2021, de https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/

8. Anexo A

Anexo A

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
I.C.A.I.

PROYECTOS FIN DE GRADO
CURSO: 4º GITT

Ficha de proyecto fin de grado
(RELLENAR CON LETRAS DE IMPRENTA EN ORDENADOR)

Titulación y optatividad: Grado en Ingeniería en Tecnologías de Telecomunicación

Alumno 1º Apellido: López
2º Apellido: Soto
Nombre: Ignacio

Teléfono de contacto: +1 (217) 695 - 6919 e-mail:
201802973@alu.comillas.edu

Título del Proyecto Fin de Grado: Aplicación de técnicas de aprendizaje automático para evaluar y predecir la actividad geomagnética solar en las comunicaciones

Director (nombre y dos apellidos): Miguel Ángel Sanz Bobi Teléfono de contacto:
e-mail: masanz@comillas.edu

Breve descripción del proyecto (5 o 6 líneas)

Según la definición de la NASA, "la actividad en la superficie del Sol crea un tipo de clima llamado clima espacial". El Sol está realmente lejos, a unos 93 millones de millas (150 millones de kilómetros), de la Tierra. Sin embargo, el clima espacial puede afectar a la Tierra y al resto del sistema solar. En el peor de los casos, ¡incluso puede dañar satélites y provocar apagones eléctricos en la Tierra! (https:/ / spaceplace.nasa.gov/ spaceweather/ en/). Este proyecto de investigación explorará la posible aplicación de técnicas de aprendizaje automático como, por ejemplo, redes neuronales convolucionales para modelar algunas características del clima espacial como es la actividad geomagnética y su impacto en las comunicaciones.

El documento final del proyecto será subido al Repositorio Institucional de Comillas con acceso público. El alumno podrá solicitar un nivel restringido de acceso (incluido el ‘‘cerrado’’ o ‘‘confidencial’’) que podrá concederse, excepcionalmente, si está plenamente justificado.

The final report of the Project will be uploaded to the Comillas Institutional Repository with public access. The student will be able to ask for a restricted access (even ‘‘closed’’ or ‘‘confidential’’) which will be exceptionally accepted if it is fully justified.

Aceptación del Director (firma y fecha)

Firmado por SANZ BOBI MIGUEL ANGEL - 25965998X el día 22/09/2021 con un certificado emitido por AC FNMT Usuarios

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

Los ODS (Objetivos de Desarrollo Sostenible), son objetivos adoptados por los Estados Miembros de la ONU en 2015 como un llamado universal para poner fin a la pobreza, proteger el planeta y garantizar el bienestar de todas las personas para 2030 [23]. Este proyecto se alinea con algunos de estos objetivos, cuyas definiciones se han extraído de [24].

- Objetivo 7: Garantizar el acceso a una energía asequible, segura, sostenible y moderna
- Objetivo 9: Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación.

El proyecto está mayoritariamente vinculado al objetivo número nueve, el fomento de la innovación tecnológica. La investigación de modelos de predicción basados en redes neuronales se puede aplicar a cualquier sector, tal y cómo lo hemos aplicado en este trabajo para la predicción del índice *Kp*. Mediante este proyecto, se busca cumplir la meta "Aumentar la investigación científica y mejorar la capacidad tecnológica de los sectores industriales de todos los países, en particular los países en desarrollo, entre otras cosas fomentando la innovación y aumentando considerablemente, de aquí a 2030, el número de personas que trabajan en investigación y desarrollo por millón de habitantes y los gastos de los sectores público y privado en investigación y desarrollo."

Por otro lado, el proyecto está alineado con el objetivo de desarrollo número 7, definido anteriormente. Con el desarrollo de una aplicación que detecte tormentas solares, conseguimos preparar las telecomunicaciones antes de que sean afectadas. Por ello, el proyecto busca cumplir con la meta "De aquí a 2030, ampliar la infraestructura y mejorar la tecnología para prestar servicios energéticos modernos y sostenibles para todos en los países en desarrollo, en particular los países menos adelantados, los pequeños Estados insulares en desarrollo y los países en desarrollo sin litoral, en consonancia con sus respectivos programas de apoyo".

ANEXO II: LISTADO Y ENLACE A LOS ARCHIVOS

Enlace a los archivos: https://drive.google.com/drive/folders/19XrOm5M-3T7l9fStvAUeMmWAejZN2tzg?usp=sharing

Listado de archivos:

- Carpeta "data": contiene los archivos importados con datos del viento solar, el campo magnético y el índice *Kp* desde enero de 2015 hasta diciembre de 2021. Contiene los siguientes tipos de archivos:
    - yyyymm_ace_mag_1h.txt : dónde yyyy representa el año y mm el mes en el que se midieron los datos sobre el campo magnético.
    - yyyymm_ace_swepam_1h.txt : dónde yyyy representa el año y mm el mes en el que se midieron los datos sobre el viento solar.
    - Kp_ap_AP_SN_F107_yyyy.txt : dónde yyyy representa el año en el que se midieron los datos del índice *Kp*.
- data.csv : archivo con todos los datos de no tormentas agrupados.
- dataStorm.csv: archivo con todos los datos de tormentas agrupados
- dataTotal.csv : archivo con todos los datos agrupados
- Kp.csv : archivo con todos los datos del índice *Kp*.
- mag.csv : archivo con todos los datos del campo magnético.
- solarWind.csv : archivo con todos los datos de la densidad de protones y su velocidad de flujo.
- KpDataConverter.ipynb : notebook que lee los archivos con los datos de *Kp* y crea Kp.csv
- MagneticFieldDataConverter.ipynb : notebook que lee los archivos con datos del campo magnético y crea mag.csv
- solarWindDataConverter.ipynb : notebook que lee los archivos con datos del viento solar y crea solarWind.csv
- DataUnifyer.ipyng : notebook que lee Kp.csv, mag.csv y solarWind.csv, interpola los valores que faltan y crea data.csv, dataStorm.csv y dataTotal.csv.
- ModeloLSTM.ipynb : notebook que entrena una red LSTM, genera los gráficos correspondientes para su análisis y lo evalúa con datos de tormenta.
- ModeloConv-MASB.ipynb : notebook que entrena una red convolucional, genera los gráficos correspondientes para su análisis y lo evalúa con datos de tormentas.
- LSTM3020.json : contiene el modelo LSTM optimizado con 30 neuronas en la primera capa y 20 en la segunda en formato json.
- LSTM3020.h5 : contiene los pesos del modelo modelo LSTM optimizado con 30 neuronas en la primera capa y 20 en la segunda.
- conv30-20-4.h5 : contiene los pesos del modelo convolucional optimizado con 30 neuronas en la primera capa, 20 en la segunda y 4 en la tercera.
- Carpeta conv30-20-4.h5py : contiene la red convolucional optimizada con 30 neuronas en la primera capa, 20 en la segunda y 4 en la tercera, con su estado guardado después de ser entrenado.
- DetectorAnomalias.ipynb : notebook de la aplicación que detecta tormentas para todas las fechas desde 2015 hasta 2021.