INSTITUTO TECNOLÓGICO DE BUENOS AIRES – ITBA

ESCUELA DE POSTGRADO

Modelo de Pronóstico como soporte para el Mantenimiento Predictivo en Filtros de Aire de una Turbina de Gas

AUTOR: Carrizo, Javier Jorge Andrés (Leg. Nº 104158)

TUTORA: Gómez, Leticia Irene

TRABAJO FINAL PRESENTADO PARA LA OBTENCIÓN DEL TÍTULO DE ESPECIALISTA EN CIENCIA DE DATOS

SAN MARTIN DE LOS ANDES, NEUQUEN

SEGUNDO CUATRIMESTRE, 2022

Índice de Contenidos

1. Introducción

"Los datos son el petróleo del siglo XXI. El despliegue de sensores y el incremento de la capacidad del procesamiento son claves en la transformación de muchos sectores y en la creación de un mundo más medible y programable" [1]

Cesar Alierta

Es posible definir tres tipos de mantenimientos en equipos industriales. *Correctivo*, que consiste en la reparación del equipo o componente luego de una falla; *preventivo*, que permite mediante inspecciones, reparaciones o reemplazo de piezas evitar posibles fallas en el equipo; y *predictivo*, que comparte con el preventivo su accionar previo a la falla, pero permitiendo a su vez determinar cuándo resulta conveniente realizar una actividad en el equipo a partir de la recolección de datos, evitando actividades innecesarias, maximizando el tiempo de vida del equipo y minimizando el riesgo de falla. [2]

En la corriente década, y catalizado por la digitalización de procesos industriales dentro del marco de proyectos de *Industry 4.0* e *Internet of Things* (*IoT)* [3], se perfila un escenario óptimo para la aplicación de técnicas de *Big Data* [4], logrando la adquisición de mediciones desde múltiples sensores que son almacenadas y permiten su análisis en tiempo real, maximizando la potencialidad del mantenimiento predictivo.

Diversos tipos de sensores existen para la obtención de datos desde los equipos industriales, los que podemos clasificar como *Process Sensors*, que miden temperaturas, presiones, caudales, etc., y a partir de informar las condiciones operativas del proceso, permiten identificar también potenciales problemas de funcionamiento en los equipos; un segundo grupo de Sensores, son los denominados *Test Sensors*, que permiten obtener mediciones como vibraciones en equipos (por ejemplo, a través de acelerómetros), mediciones acústicas, señales eléctricas, condiciones ambientales, etc. Estas dos primeras categorías de sensores no intervienen en el comportamiento del proceso, es decir, tienen una postura pasiva; en contraste, una tercera categoría de sensores depende de señales que se envían al proceso, para obtener la medición, los que se conocen como *Test Signal* (LCSR Test, SHI, TDR, etc). [5]

Esta enorme generación de datos estructurados o no, que consideramos *Big Data*, requiere decisiones para su almacenamiento y procesamiento, con el objetivo de facilitar la aplicación de modelos de *machine learning* con el fin de predecir eventos, complementados con visualizaciones que permitan resumir la información necesaria para la toma de decisiones. En tal sentido, el *Cloud Computing* juega un rol cada vez más importante, brindando una plataforma de acceso a recursos tecnológicos por un costo menor respecto de alternativas *on-premise*. [6]

Este trabajo se enfoca en el mantenimiento predictivo, el cual se basa principalmente en los siguientes módulos metodológicos: adquisición y preprocesamiento de datos, construcción de indicadores de salud, detección y localización (tempranas) de anomalías, previsión y pronóstico del estado de salud del sistema/máquina o posibles problemas/tendencias descendentes (incluida la predicción de la vida útil restante o pronósticos de fallas) y acciones correctivas cuando se detectan o predicen estados no deseados de los sistemas.

En particular, el pronóstico (*forecasting*), se refiere a posibles problemas que surjan en el futuro (próximo). El modelo de *forecasting* que se implemente debe ser lo suficientemente preciso para ofrecer predicciones confiables, mientras que el tiempo de reacción debe ser tal que permita realizar intervenciones significativas en el proceso o en el tiempo de ejecución de las máquinas antes de que se rompan algunos componentes/piezas. [7]

2. Contexto

En la actualidad, los mantenimientos correctivo y preventivo en la industria se encuentran aplicados masivamente [8], sin embargo, el mantenimiento predictivo aún se encuentra evolucionando, con un menor grado de aplicación o en el mejor de los casos con una aplicación parcial por limitaciones de infraestructura tecnológicas de *hardware* y *software*.

Otras actividades explotan de manera consistente técnicas de *big data* y *machine learning*, como telecomunicaciones, *e-commerce* y *fintech* entre otras.

Por otro lado, se observa el abaratamiento de sensores y elementos de medición para la adquisición de datos; junto a una también reducción de costos en infraestructura para la comunicación de los datos obtenidos, y no en menor medida, ocurre algo similar en relación con su almacenamiento, con las opciones disponibles en la nube o *cloud computing.* Este nuevo contexto facilita procesos de digitalización.

Una compañía de generación eléctrica de ciclo combinado posee numerosos equipos de alta criticidad que deben mantenerse en servicio el mayor tiempo posible. En la Figura 1 podemos ver un esquema del funcionamiento de una central de ciclo combinado.

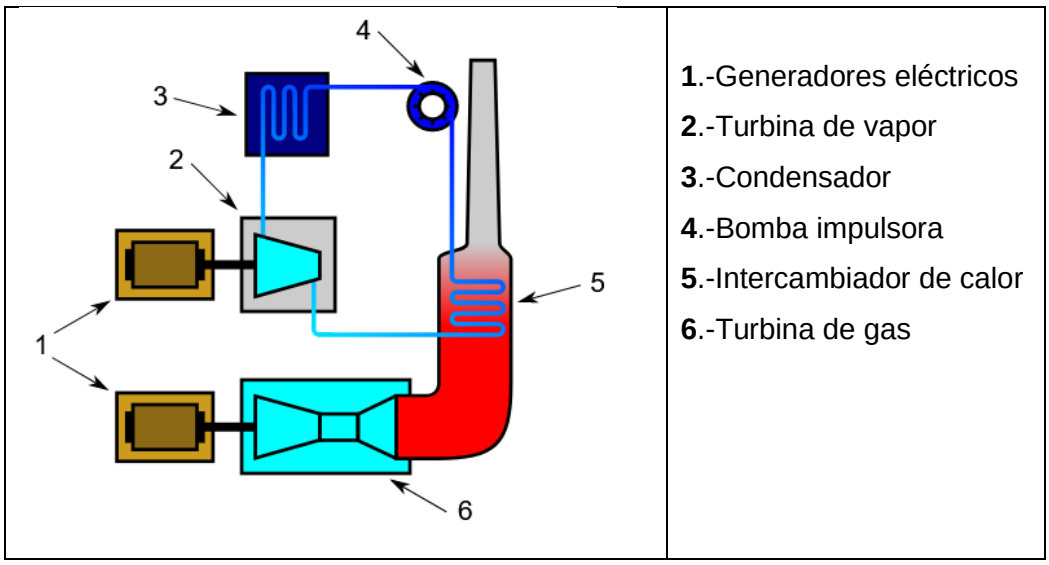

Figura 1 - Esquema del funcionamiento de una central de ciclo combinado
Fuente: https://commons.wikimedia.org/wiki/File:COGES_diagram.svg

El funcionamiento de una turbina de gas, por su diseño básico, requiere que ingiera grandes cantidades de aire. La filtración se aplica al aire de entrada para brindar protección contra los efectos del aire contaminado. Los diferentes tipos de contaminantes en el aire pueden causar diversos problemas que impactan negativamente en la confiabilidad, disponibilidad y tiempo entre revisiones de los componentes internos de las turbinas de gas. El propósito principal de la filtración de entrada es limpiar el aire para cumplir con los objetivos operativos de la máquina [9]. En la Figura 2 se puede ver un esquema de Filtro de Aires y Turbina de Gas.

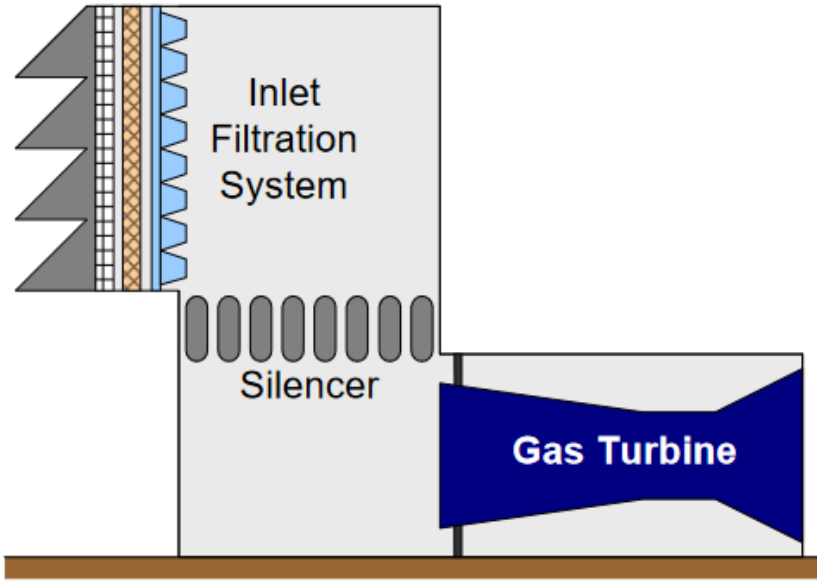

Figura 2 - Esquema de Filtro de Aires y Turbina de Gas
Fuente: Gas Machinery Research Council Southwest Research Institute, Guideline for Gas Turbine Inlet Air Filtration Systems

3. Presentación del Problema

Es necesario contar con una predicción que permita anticiparse a una contaminación en los filtros de entrada aire a la turbina, permitiendo planificar el cambio de los mismos, sin tener que recurrir a la detención intempestiva de la turbina que está operando.

4. Justificación del Estudio

Como se mencionó en la sección 2, garantizar la calidad del aire de entrada a la turbina de gas tiene una importancia superlativa en el funcionamiento del equipo. Poder anticipar una disminución de la calidad del aire permitiría actuar de manera proactiva evitando paradas no programadas en la operación de la turbina.

El cambio de filtros puede demorar entre tres días y una semana, dependiendo si se cambian parcialmente <pre-filtro> o completo, incluyendo el filtro fino. El pre-filtro puede reemplazarse con la turbina operando, no así el filtro fino.

5. Alcances del Trabajo y Limitaciones

El presente trabajo, se restringirá en particular al sistema de filtración de aire de las turbinas de gas, para la que se presentará una experiencia especifica de *forecasting* aplicada al sistema de filtros de aire mencionados, sin embargo, puede servir de caso testigo para replicarla en otros equipos industriales y otras variables relevadas donde agregue valor predecir su comportamiento.

6. Objetivos

6.1. Objetivo General

Diseñar e implementar un modelo de *forecasting* que permita predecir la calidad del aire que ingresa a la turbina de gas con la anticipación suficiente para poder actuar ante desvíos que afecten el funcionamiento de la turbina.

6.2. Objetivos Específicos

- Identificar la variable medida que mejor explique la calidad del aire en filtros para intentar predecirla.
- Identificar una o más covariables que aporten información para lograr una mejor predicción.
- Elegir un modelo apropiado para realizar la predicción.
- Almacenar las predicciones con el fin de evaluar y validar el modelo, permitiendo detectar cuando reentrenarlo.
- Calcular intervalos de la predicción para el modelo elegido.
- Acompañar las predicciones obtenidas con una visualización que facilite su interpretación.

7. Metodología Aplicada

Se llevará a cabo el caso de estudio con una aplicación en tiempo real de un modelo de *forecasting* que predecirá el diferencial de presión entre la salida y entrada de filtro de aire de una turbina de gas, variable que permite conocer la calidad del aire que ingresa a la turbina.

7.1. Modelo Conceptual

El modelo conceptual seguirá las siguientes etapas de implementación que se muestran en la Figura 3:

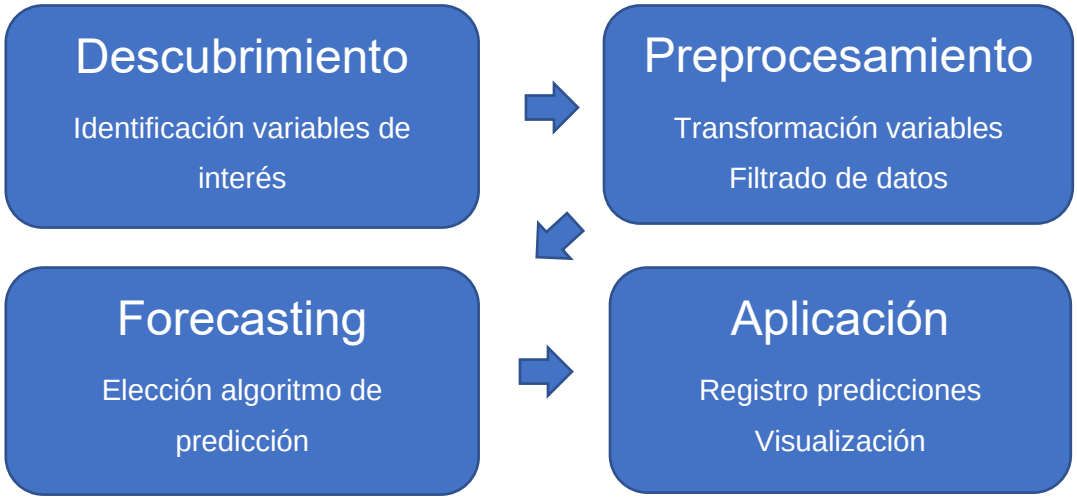

Figura 3 – Etapas de implementación

7.2. Algoritmos Utilizados

Se utilizarán dos modelos predictivos basados en redes neuronales recurrentes, dado que los parámetros de modelos clásicos como ARIMA[1] o similares, no presentarían buenas estimaciones al no tratarse de series estacionarias.

[1] *Auto Regressive Integrated Moving Average*

Luego de entrenados y testeados, se analizará la eficacia de cada modelo, y se los comparará entre sí, para determinar el que mejor se ajuste a la variable que se desea predecir.

El primer algoritmo a utilizar será una red neuronal recurrente N-BEATS incluyendo covariables. Se opta por una red neuronal considerando que la serie no es estacionaria, por lo que no se recomienda recurrir a algoritmos.

N-BEATS es un tipo de red neuronal que se describió por primera vez en un artículo de 2019 [10] . Los autores informaron que N-BEATS superó al ganador de la competencia de pronóstico M4[2] en un 3 %. El ganador de M4 fue un híbrido entre una red neuronal recurrente y el suavizado exponencial de *Holt-Winters*, mientras que N-BEATS implementa una arquitectura neuronal profunda "pura".

El modelo N-BEATS consta de una secuencia de *stacks*, cada una de las cuales combina varios bloques. Los bloques conectan redes *feedforward* a través de enlaces de predicción y *backcast*. Un bloque “elimina la porción de la señal y luego, el nuevo bloque se enfoca en el error residual que los bloques anteriores no pudieron desentrañar. Cada bloque genera un pronóstico parcial, con su enfoque puesto en las características locales de la serie temporal. Cada *stack* agrega los pronósticos parciales a través de los bloques que comprende y luego pasa el resultado al siguiente *stack*. El propósito de los *stacks* consiste en identificar patrones no locales a lo largo del periodo de tiempo completo "mirando hacia atrás". Finalmente, los pronósticos parciales se juntan en un pronóstico global al nivel del modelo. Ver Figura 4.

N-BEATS toma como sus hiperparámetros:

- Los tamaños de las capas de entrada y salida
- El número de bloques por stack
- La cantidad de nodos
- El tamaño del lote
- La cantidad de iteraciones

El producto de estos hiperparámetros define el tamaño del tensor del modelo.

[2] *https://mofc.unic.ac.cy/m4/ M Open Forecasting Center (MOFC)*

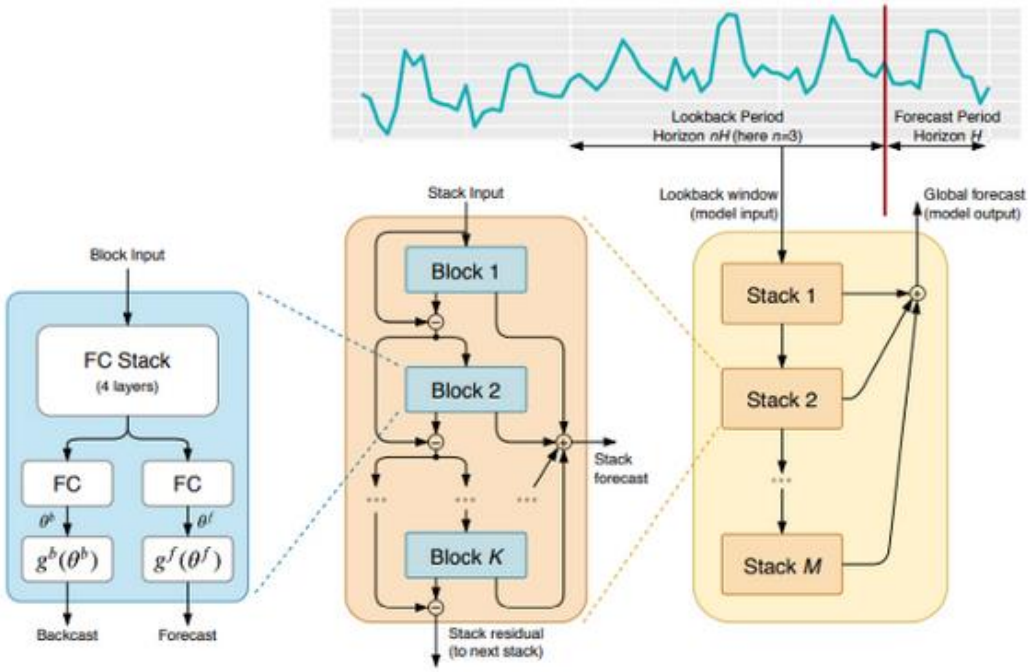

Figura 4 - Esquema del modelo de red neuronal N-BEATS

Fuente: https://arxiv.org/pdf/1905.10437.pdf, "N-BEATS", Oreshkin and Carpov (2020)

Este modelo acepta un argumento de covariable, observando solo los valores pasados de la serie al hacer una predicción. Ver Figura 5.

Una covariable es una serie de tiempo que puede ayudar en el pronóstico de la serie objetivo, pero que no nos interesa pronosticar. Las covariables pasadas, son series temporales cuyos valores pasados se conocen en el momento de realizar la predicción. [11]

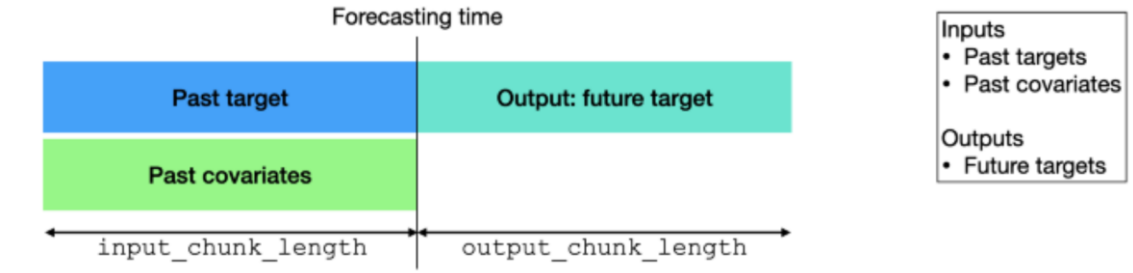

Figura 5 - Representación de modelos de covariables pasadas

Fuente: https://unit8.com/resources/time-series-forecasting-using-past-and-future-external-data-with-darts/

Complementariamente al algoritmo N-BEATS mencionado, se evaluó otro algoritmo de Redes Neuronales Recurrentes en su variante LSTM[3] [12] que también soporta covariables pasadas.

LSTM se trata de un modelo de red neuronal que utiliza un *encoder* para codificar fragmentos de entrada de longitud fija y una red completamente conectada para producir salidas de longitud fija.

En particular, la variante LSTM introduce bucles automáticos para producir caminos en los que el gradiente puede fluir durante períodos prolongados. El peso de este bucle automático está condicionado por el contexto, en lugar de permanecer constante. En la Figura 6, se observa que las celdas se conectan de forma recurrente entre sí, reemplazando las unidades ocultas habituales de las redes recurrentes ordinarias.

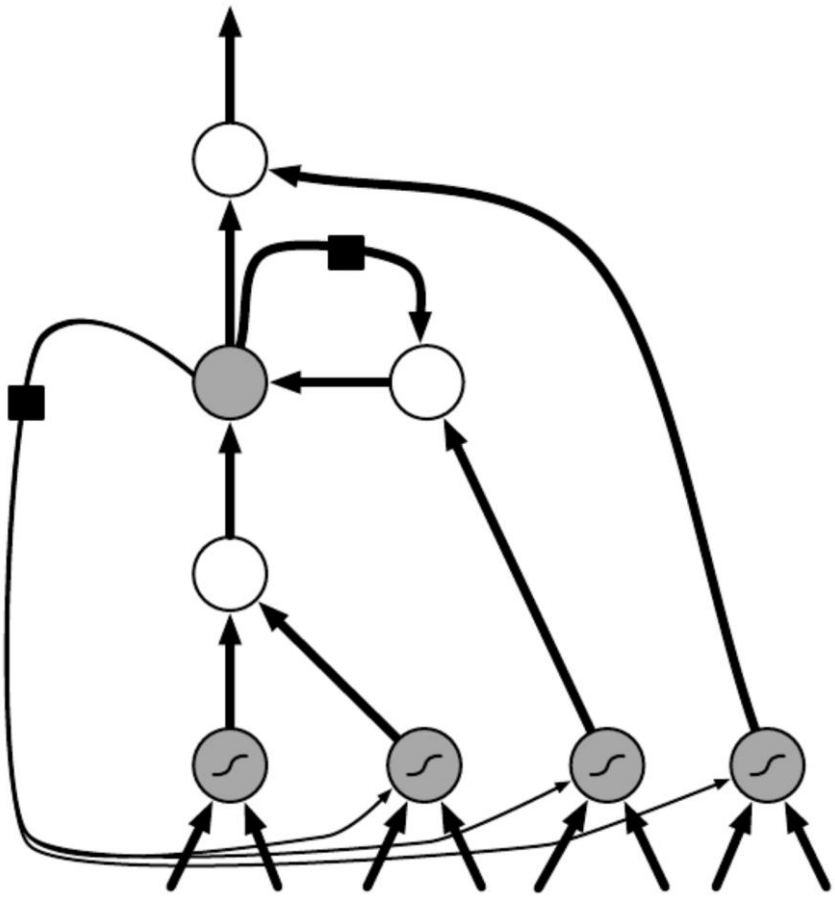

Figura 6 - Esquema del modelo de red neuronal LSTM

Fuente: Deep Learning, Ian Goodfellow, Yoshua Bengio and Aaron Courville

[3] *Long Short-Term Memory*

8. Desarrollo del Modelo

8.1. Fuentes de Información

Los datos provienen de una aplicación que recolecta los datos y eventos en tiempo real, mediante interfaces con sensores[4] instalados en la turbina de gas y sus equipos satélites.

En particular para el presente trabajo, se utilizarán las siguientes variables:

- Diferencial de Presión de Filtros *(Variable a Predecir)*
- Tiempo de operación del IGV[5] *(Covariable)*

8.2. Proceso de ETL Aplicado

Para integrar la información y entrenar el modelo se programó un ETL, en el mismo software de recolección que permite la aplicación de reglas, que ejecuta las siguientes acciones:

La variable diferencial de presión de filtros se obtiene directamente del sensor, que a partir de las lecturas de presión a la salida y a la entrada de los filtros calcula las diferencias entre cada medición; luego en la aplicación se estableció una regla para obtener el promedio diario en una nueva variable.

Para la covariable tiempo de operación del IGV, se calcula un acumulado diario en cantidad de días a partir del último cambio de filtros. Cada valor diario obtenido acumula entre un mínimo de 0 (cero) si no estuvo en funcionamiento y un máximo de 1 (uno) si operó durante las 24 hs.

Cabe mencionar que los datos son previamente filtrados a partir de otra variable categórica denominada estado de carga, tomando los valores para variable y covariable exclusivamente cuando el Estado de Carga toma el valor “En Menor Performance”, que

[4] *Process Sensor*
[5] *"Inlet Guide Vane" (IGV)* - Álabe de guía de entrada

se corresponde con la situación de operación normal de la turbina de gas, es decir, cuando está entregando toda la potencia posible; mientras que se excluyen estados de mantenimiento, despacho o forzada entre otros.

8.3. Procesamiento de los Datos

Para el entrenamiento y validación del modelo, se identifica un periodo de tiempo entre cambios de filtros y se desarrolla una notebook en Python recurriendo principalmente a la librería *Darts*[6], que ofrece diferentes modelos para series temporales, tanto clásicos como ARIMA, pero también basados en redes neuronales recurrentes, como LSTM o N-BEATS.

Por otro lado, para la implementación del modelo, se toman del sistema que recoleta los datos, las últimas 20 mediciones de cada una de estas variables, junto con la fecha asociada. Esta información se transforma en un *json* que se envía al modelo publicado en una URL, obteniendo la predicción de los siguientes 10 días para la variable de interés, diferencial de filtros en este caso, que es almacenada en el mismo sistema utilizado para la recolección.

Finalmente, el mismo *script* que envía datos y recibe las predicciones diariamente, almacena estas últimas en un archivo plano, permitiendo un estudio de la incertidumbre del modelo para diferentes horizontes de predicción, es decir, poder evaluar la precisión del modelo para ***n*** de días a futuro.

[6] *https://unit8co.github.io/darts/*

8.4. Panel de Visualización

La visualización consiste en un gráfico de serie temporal que permite observar los datos reales hasta el tiempo presente y a continuación la predicción de la variable para los próximos ***n*** días solicitados, como puede observarse en la Figura 7.

El panel forma parte de un *dashboard,* desarrollado durante el presente trabajo, en la misma aplicación donde se realiza la recolección y gestión de datos.

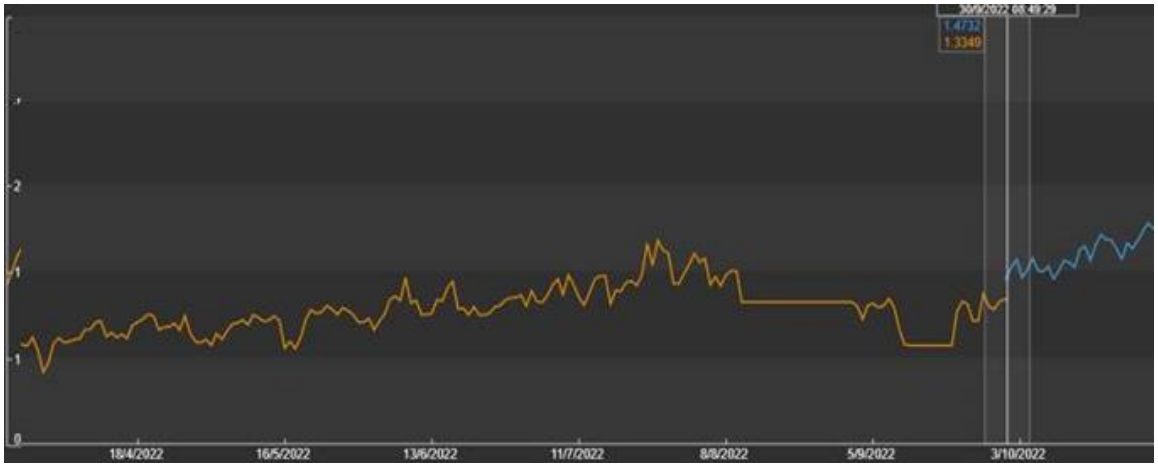

Figura 7 – Visualización de una serie temporal incluyendo valores históricos y predicción

9. Resultados Experimentales

A continuación, se describirán las diferentes etapas realizadas de acuerdo al modelo conceptual mencionado con anterioridad.

9.1. Análisis Exploratorio

Se genera un set de datos para uno de los filtros entre dos eventos de cambio, para tener un ciclo de vida completo. El set de datos contiene los valores diarios del Diferencial de Filtros y Operación de IGV, identificando los datos utilizados para entrenar el modelo (amarillo) de los empleados para su validación (verde). Ver Figura 8.

time	filtro	igva
2021-01-03	1,20	0,62
2021-01-04	1,21	1,31
2021-01-05	1,19	2,09
2021-01-06	1,23	2,96
2021-01-07	1,25	3,88
2021-01-08	1,21	4,79
2021-01-09	1,21	5,67
2021-01-10	1,28	6,67
2021-01-11	1,18	7,56
2021-01-12	1,25	8,42
2021-01-13	1,31	9,42
2021-01-14	1,34	10,42
2021-01-15	1,27	11,42
2021-01-16	1,12	12,38
2021-01-17	1,14	13,34
2021-01-18	1,28	13,92
2021-01-19	1,33	14,47
2021-01-20	1,35	15,18
2021-01-21	1,36	16,18
2021-01-22	1,33	17,18
2021-01-23	1,31	18,18
2021-01-24	1,35	19,05
2021-01-25	1,33	19,92
2021-01-26	1,29	20,83
2021-01-27	1,33	21,83
2021-01-28	1,29	22,83
2021-01-29	1,27	23,83
2021-01-30	1,26	24,83
2021-01-31	1,23	25,61
2021-02-01	1,36	26,38
2021-02-02	1,38	26,78
2021-02-03	1,41	27,66
2021-02-04	1,35	28,66
2021-02-05	1,37	29,51
2021-02-06	1,39	30,33
2021-02-07	1,38	31,03
2021-02-08	1,32	32,03
2021-02-09	1,38	33,03
2021-02-10	1,42	34,03
2021-02-11	1,42	35,03
2021-02-12	1,52	36,03
2021-02-13	1,48	36,71
2021-02-14	1,47	37,71
2021-02-15	1,40	38,71
2021-02-16	1,39	39,71
2021-02-17	1,43	40,71
2021-02-18	1,46	41,71
2021-02-19	1,38	42,71
2021-02-20	1,35	43,44
2021-02-21	1,40	44,42
2021-02-22	1,44	45,35
2021-02-23	1,42	46,08
2021-02-24	1,43	47,08
2021-02-25	1,43	48,08
2021-02-26	1,48	49,08
2021-02-27	1,53	49,90
2021-02-28	1,54	50,70
2021-03-01	1,56	51,65
2021-03-02	1,57	52,54
2021-03-03	1,61	53,54
2021-03-04	1,59	54,54
2021-03-05	1,48	55,54
2021-03-06	1,54	56,54
2021-03-07	1,57	57,52
2021-03-08	1,56	58,52
2021-03-09	1,56	59,41
2021-03-10	1,60	60,41
2021-03-11	1,60	61,41
2021-03-12	1,62	62,27
2021-03-13	1,61	63,27
2021-03-14	1,63	64,27
2021-03-15	1,62	65,23
2021-03-16	1,65	66,12
2021-03-17	1,44	67,12
2021-03-18	1,56	68,12
2021-03-19	1,65	68,95
2021-03-20	1,64	69,72
2021-03-21	1,63	70,58
2021-03-22	1,69	71,43
2021-03-23	1,72	72,19
2021-03-24	1,70	72,99
2021-03-25	1,85	73,99
2021-03-26	1,79	74,68
2021-03-27	1,76	75,66
2021-03-28	1,73	76,41
2021-03-29	1,70	77,41
2021-03-30	1,75	78,28
2021-03-31	1,78	79,28
2021-04-01	1,82	80,17
2021-04-02	1,66	81,04
2021-04-03	1,70	82,04
2021-04-04	1,76	82,09
2021-04-05	1,81	82,09
2021-04-06	1,80	82,59
2021-04-07	1,80	83,59
2021-04-08	1,85	84,59
2021-04-09	1,88	85,57

Figura 8 - Set de datos para entrenamiento y validación del Modelo

Se realiza un análisis exploratorio de la variable de interés, cuyo resultado puede verse en la Figura 9.

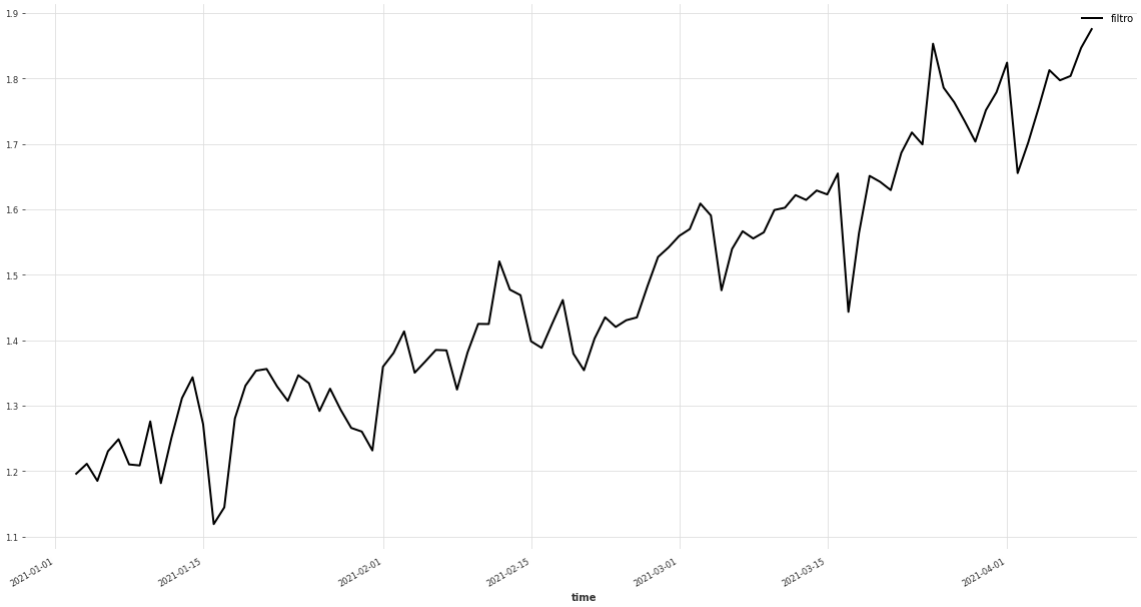

Figura 9 – Comportamiento del Diferencial de Filtros entre cambio de Filtros

Al incrementarse el valor obtenido en la variable Diferencial de Filtros, indica mayor presencia de contaminantes en los filtros, determinando el cambio de los mismos al superar el valor de alarma establecido.

Por otro lado, para ese mismo periodo de tiempo, se observa el tiempo de operación acumulado del IGV, como se muestra en la Figura 10.

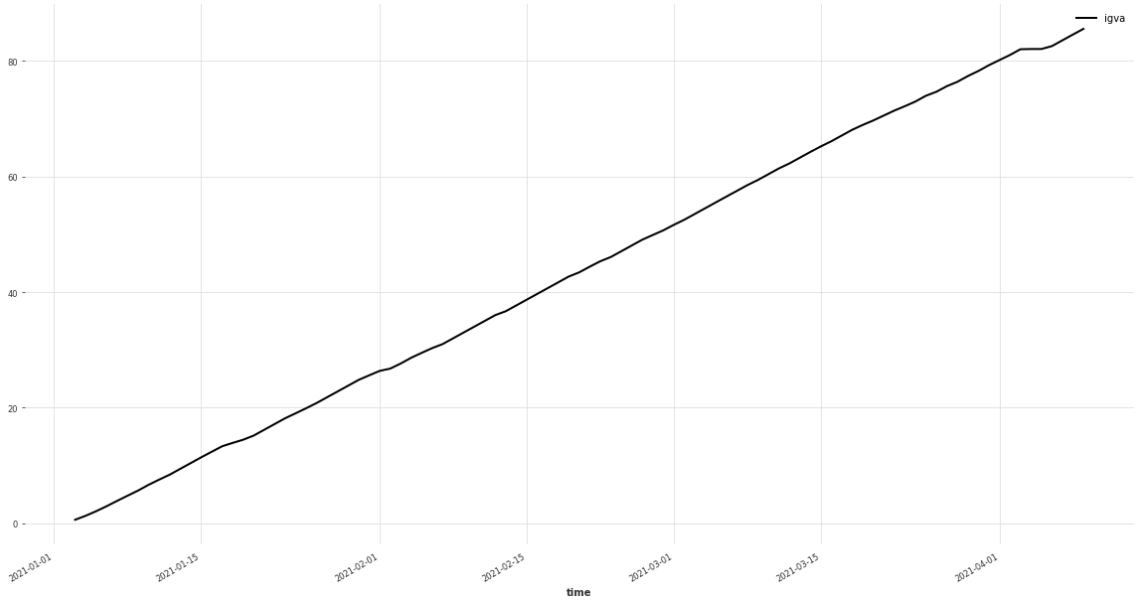

Figura 10 – Tiempo acumulado de operación IGV entre cambio de Filtros

En la Figura 11 se presentan ambas variables, diferencial de filtros y tiempo de operación IGV, normalizadas para poder compararlas.

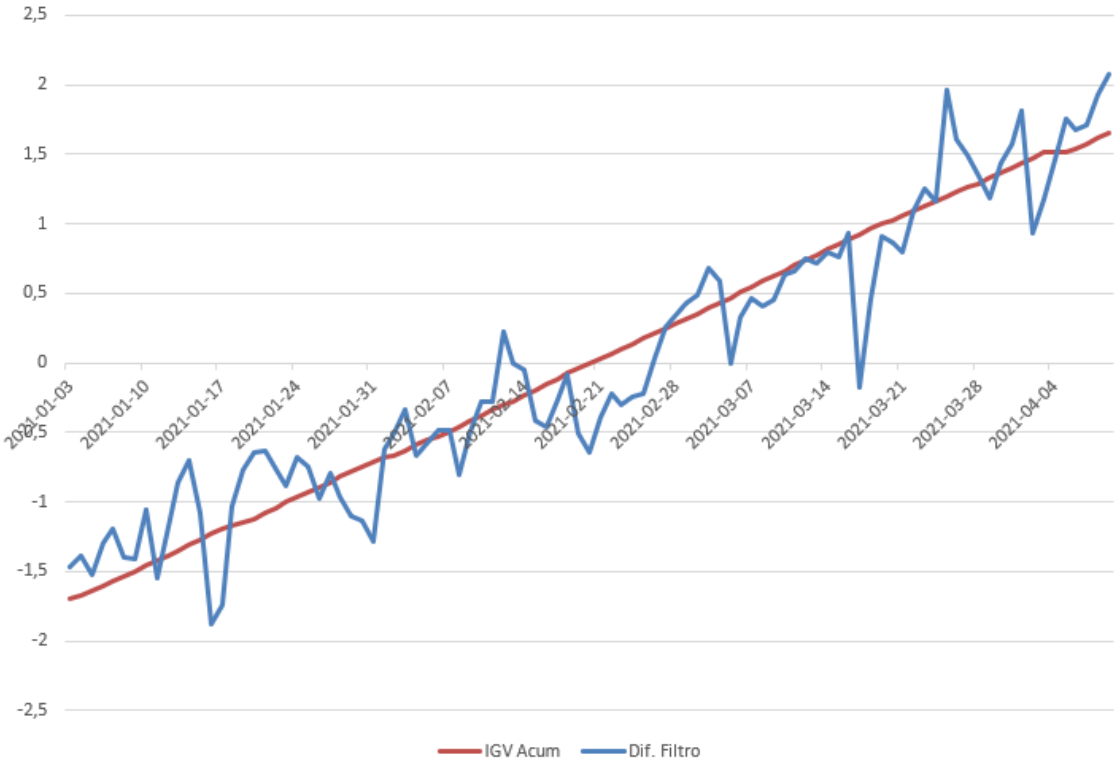

Figura 11 – Diferencial de Filtros y Tiempo acumulado de operación IGV normalizadas

Se evidencia visualmente una correlación positiva entre ambas variables, lo que justifica el uso del tiempo de operación IGV como covariable del modelo a implementar para predecir el diferencial de filtros, aportando información relevante y en consecuencia mejorar la predicción.

9.2. N-BEATS

Se entrena y valida un modelo[7] utilizando los datos de este periodo temporal, recurriendo a la red neuronal N-BEATS como se mencionó anteriormente, considerando la operación de IGV como covariable y realizando 100 interacciones durante el entrenamiento.

[7] Notebook de entrenamiento disponible en Anexo

Dado que se trata de un modelo de redes neuronales, se utiliza un *likelihood* basado en una regresión por cuantiles *Quantile Loss Metric,* para obtener el intervalo de la predicción.

Así como las regresiones minimizan la función de pérdida del error cuadrático para predecir una estimación de un solo punto, las regresiones por cuantiles minimizan la pérdida por cuantiles al predecir un determinado cuantil. La pérdida por cuantiles difiere según el cuantil evaluado, de modo que a más errores negativos se les penaliza más para los cuantiles más altos y a más errores positivos se les penaliza más para los cuantiles más bajos [13]. Ver Figura 12.

Figura 12 – Predicción del modelo propuesto junto a su intervalo

Para una predicción de 15 valores se obtiene un valor MAPE = 2.66. Dada una serie temporal de valores reales y una serie temporal de valores predichos ambos de determinada longitud, el MAPE[8] o Error porcentual absoluto medio, es un valor porcentual calculado como:

$$100 \cdot \frac{1}{T} \sum_{t=1}^{T} \left| \frac{y_t - \hat{y}_t}{y_t} \right|.$$

El error porcentual absoluto medio (MAPE) expresa la exactitud como un porcentaje del error. Debido a que el MAPE es un porcentaje, puede ser más fácil de entender que

[8] *Mean Absolute Percentage Error* [14]

otros estadísticos de medición de exactitud. Por ejemplo, si el MAPE es 5, en promedio, el pronóstico está errado en un 5%.

En la Figura 13, podemos observar los pronósticos históricos que habría producido el modelo con una ventana de entrenamiento en expansión y un horizonte de pronóstico de 10 días, obteniendo un $R^2 = 0.8$ entre la medición real histórica y los pronósticos que se hubieran obtenido en cada caso.

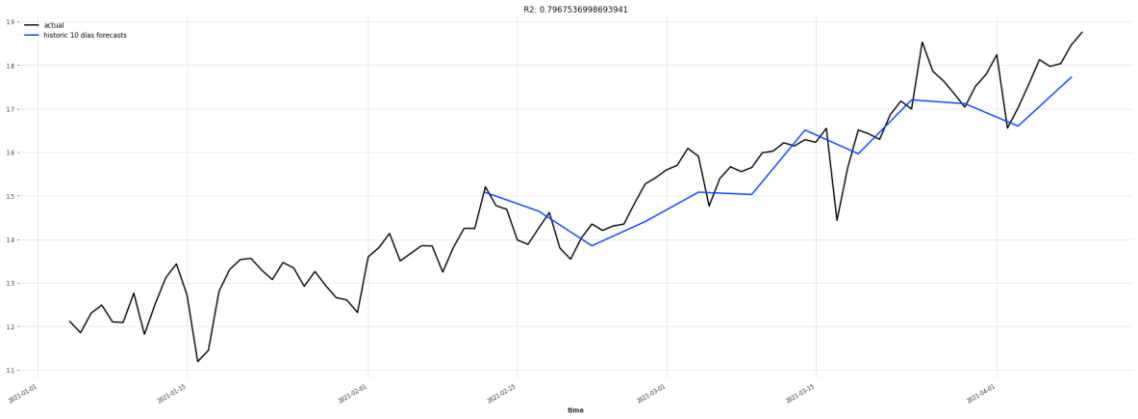

Figura 13 – Predicción histórica del modelo para un horizonte de 10 días

9.3. LSTM

Complementariamente, se entrena y valida un segundo modelo[9] utilizando los datos del mismo periodo temporal, recurriendo al algoritmo LSTM como se mencionó anteriormente, considerando la operación de IGV como covariable y realizando 100 interacciones durante el entrenamiento.

En este caso, para una predicción a 15 días, es decir, el mismo periodo evaluado en el modelo de N-BEATS, se obtiene un MAPE = 7.28.

En la Figura 14 se observa la predicción obtenida, en este caso sin incluir el intervalo dado que se trata de un modelo determinístico, a diferencia del anterior que era

[9] Incluido en la Notebook de entrenamiento disponible en Anexo

estocástico o probabilístico, por lo que para la misma entrada de datos invariablemente se obtendrá la misma salida.

Figura 14 – Predicción del modelo alternativo recurriendo a LSTM

10. Conclusiones y Trabajo a Futuro

Como consecuencia de los resultados experimentales, se decide optar por el modelo que recurre al algoritmo N-BEATS por presentar una mejor predicción respecto LSTM, dado que se obtuvo un menor MAPE (2.66 N-BEATS Vs. 7.28 LSTM).

Con el presente trabajo se logró de manera simple y efectiva obtener una predicción que permite anticiparse a una contaminación en los filtros de entrada aire a la turbina, permitiendo planificar el cambio de los mismos, sin tener que recurrir a la detención intempestiva de la turbina que está operando.

Como trabajo futuro, se implementará un indicador de salud del equipo aplicando PCA[10], análisis de componentes principales, consolidando las múltiples variables medidas en los filtros, permitiendo complementar la predicción de la serie temporal del diferencial de filtros aquí presentada.

[10] *Principal Components Analisis*

Sobre la o las componentes principales obtenidas que resulten relevantes, se aplicará un Control Estadístico de Proceso que permita detectar desvíos anormales y en consecuencia un análisis de causa raíz y potencial plan de acción correctiva si la investigación realizada lo justifica.

De esta manera se alcanzará un monitoreo exhaustivo del equipo que contemple tres aspectos del mantenimiento predictivo.

- Modelo predictivo de las variables críticas para anticiparse a desvíos que requieran un mantenimiento correctivo
- PCA para obtener un indicador de la salud del equipo considerando todas las variables observadas
- Control estadístico de procesos sobre variables críticas y componentes principales, como metodología para la detección de anomalías.

Este modelo propuesto para el mantenimiento predictivo será extendido a otros equipos críticos.

11. Referencias

[1] César Alierta, presidente de Telefónica, en la Real Academia de Ciencias Económicas y Financieras (Racef), marzo 2022.

[2] S. Wu, N. Gebraeel, M. A. Lawley and Y. Yih, *"A Neural Network Integrated Decision Support System for Condition-Based Optimal Predictive Maintenance Policy," in IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, no. 2, pp. 226-236, March 2007.

[3] GILCHRIST A., *"Industry 4.0: The Industrial Internet of Things".* Bangken, Nonthaburi, Thailandm, Springer Nature, 2016.

[4] GANDOMI A. and HAIDER M. *"Beyond the hype: Big data concepts, methods, and analytics"*, International Journal of Information Management, 2004.

[5] H. M. Hashemian, *"State-of-the-Art Predictive Maintenance Techniques," in IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 1, pp. 226-236, Jan. 2011.

[6] S. Khare and M. Totaro, "Big Data in IoT," *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kanpur, India, 2019.

[7] Edwin Lughofer and Moamar Sayed-Mouchaweh, *Predictive Maintenance in Dynamic Systems Advanced Methods, Decision Support Tools and Real-World Applications,* Springer Nature Switzerland AG, 2019.

[8] Behnam Emami-Mehrgani, W. Patrick Neumann, Sylvie Nadeau and Majid Bazrafshan, *Considering human error in optimizing production and corrective and preventive maintenance policies for manufacturing systems, Applied Mathematical Modelling* (2015).

[9] *Gas Machinery Research Council Southwest Research Institute, Guideline for Gas Turbine Inlet Air Filtration Systems*, April 2010.

[10] Boris N. Oreshkin and Dmitri Carpov. *N-BEATS: Neural Basis Expansion Analysis For Interpretable Time Series Forecasting*, Published as a conference paper at ICLR 2020.

[11] Julien Herzen, *Time Series Forecasting Using Past and Future External Data with Darts.*

[12] Ian Goodfellow, Yoshua Bengio and Aaron Courville*, Deep Learning*, MIT Press, vol 10.10, pp. 411-415, 2016.

[13] Max Ghenis, 2018 *Quantile regression, from linear models to trees to deep learning.*

[14] Myttenaere, B Golden, B Le Grand and F Rossi. *"Mean absolute percentage error for regression models"*, Neurocomputing 2016.

12. Anexo

12.1. Notebook para entrenamiento de modelos

```
# Import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from darts import TimeSeries
from darts.models import NBEATSModel, BlockRNNModel
from darts.utils.likelihood_models import QuantileRegression
from darts.dataprocessing.transformers import Scaler, MissingValuesFiller
from darts.metrics import rho_risk
from darts.metrics import mape, r2_score


# Create timeseries object for filtro
df = pd.read_excel('c:\DataSet.xlsx')
series = TimeSeries.from_dataframe(df, 'time', 'filtro')
series


#Create timeseries object for igv
seriesigv = TimeSeries.from_dataframe(df, 'time', 'igva')
seriesigv


# Create training and validation sets
train, val = series.split_after(pd.Timestamp("20210330"))


# Training n-Beats model
filtro_model = NBEATSModel(
    input_chunk_length=20,
    output_chunk_length=10,
    random_state=0,
    n_epochs=100,
    likelihood=QuantileRegression([0.05, 0.1, 0.5, 0.9, 0.95]),
)

filtro_model.fit(
      train,
      past_covariates=seriesigv,
      verbose=True
)
```

```
# Predict Using Quantile Loss
pred = filtro_model.predict(
      n=15,
      num_samples=30,
      past_covariates=seriesigv
)

plt.figure(figsize=(30, 10))

series.plot(label="actual (filtro)")
pred.plot(label="forecast (filtro)")

print("MAPE of median forecast: %.2f" % mape(series, pred))

for rho in [0.05, 0.1, 0.5, 0.9, 0.95]:
    rr = rho_risk(series, pred, rho=rho)
    print("rho-risk at quantile %.2f: %.2f" % (rho, rr))


# Historical Forecasts
def display_forecast(pred_series,
      ts_transformed,
      forecast_type,
      start_date=None):

    plt.figure(figsize=(30, 10))
    if start_date:
        ts_transformed = ts_transformed.drop_before(start_date)
    ts_transformed.univariate_component(0).plot(label="actual")
    pred_series.plot(label=("historic " + forecast_type + " forecasts"))
    plt.title(
        "R2: {}".format(r2_score(
      ts_transformed.univariate_component(0), pred_series))
    )
    plt.legend()

pred_series = filtro_model.historical_forecasts(
    series,
    start=pd.Timestamp("20210203"),
    forecast_horizon=10,
    stride=5,
    retrain=False,
    verbose=True,
)
```

```
display_forecast(
      pred_series,
      series,
      "10 días",
      start_date=pd.Timestamp("20210103")
)

# Save model
filtro_model.save_model('c:/filtro_model_cov.pth.tar')




# Training LSTM model
filtro_model_2 = BlockRNNModel(
    model="LSTM",
    dropout=0,
    batch_size=16,
    n_epochs=100,
    optimizer_kwargs={"lr": 1e-3},
    model_name="Filtros LSTM",
    log_tensorboard=True,
    random_state=42,
    input_chunk_length=20,
    output_chunk_length=10,
)

filtro_model_2.fit(
    train,
    past_covariates=seriesigv,
    verbose=True,
)


# Using Derministic Predict

pred = filtro_model_2.predict(n=15, num_samples=1,
past_covariates=seriesigv)

plt.figure(figsize=(30, 10))

series.plot(label="actual (filtro)")
pred.plot(label="forecast (filtro)")

print("MAPE of median forecast: %.2f" % mape(series, pred))
```