

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

**Análisis Espectral y Aplicación de
Aprendizaje Automático para la
Predicción y Detección de Anomalías en
Sistemas Energéticos**

Alex Javier Perez Vilañez

Ingeniería en Ciencias de la Computación

Trabajo de fin de carrera presentado como requisito para la
obtención del título de Ingeniero en Ciencias de la Computación

Quito, 19 de Diciembre de 2024

**UNIVERSIDAD SAN FRANCISCO DE QUITO
USFQ**

Colegio de Ciencias e Ingeniería

**HOJA DE CALIFICACIÓN DE TRABAJO DE FIN DE
CARRERA**

**Análisis Espectral y Aplicación de Aprendizaje
Automático para la Predicción y Detección de Anomalías
en Sistemas Energéticos**

Alex Javier Perez Vilañez

Nombre del profesor, Título académico:

Felipe Grijalva, Ph.D.

Quito, 19 de Diciembre de 2024

© Derechos de Autor

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en la Ley Orgánica de Educación Superior del Ecuador.

Nombres y apellidos:	Alex Javier Perez Vilañez
Código:	00329229
Cédula de Identidad:	1726215799
Lugar y fecha:	Quito, 19 de Diciembre de 2024

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en <http://bit.ly/COPETheses>.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on <http://bit.ly/COPETheses>.

Para Belén, Daniela y María Fernanda, por ser luz en medio de la penumbra.

Resumen

La transición hacia fuentes de energía renovable ha aumentado significativamente debido a la preocupación por la sostenibilidad y el cambio climático. Sin embargo, los sistemas energéticos modernos son vulnerables a ataques cibernéticos, como los ataques de inyección de datos falsos (FDIA), que pueden alterar los patrones de generación y demanda, afectando la estabilidad del sistema. Este proyecto tiene como objetivo fortalecer la ciberseguridad de los sistemas energéticos mediante modelos predictivos como LSTM, TCN, transformers y un modelo híbrido TCN+LSTM, evaluados en configuraciones SISO y MIMO. Además, se emplearon técnicas de detección de anomalías como Isolation Forest, LOF, Elliptic Envelope y One-Class SVM, evaluadas con métricas como AUC, F1-Score y matrices de confusión, considerando diferentes niveles de ruido y contaminación. Los resultados destacan al modelo LSTM como el más robusto para patrones temporales, mientras que Elliptic Envelope sobresale en la detección de anomalías en datos originales, y los transformers son efectivos con datos aumentados, mejorando la resiliencia del sistema frente a amenazas cibernéticas.

Palabras clave: SISO, MIMO, FDIA, LSTM, TCN, Transformers, Elliptic Envelope, Anomalías.

Abstract

The transition to renewable energy sources has significantly increased due to concerns about sustainability and climate change. However, modern energy systems are vulnerable to cyber-attacks, such as False Data Injection Attacks (FDIA), which can alter generation and demand patterns, affecting system stability. This project aims to enhance energy system cybersecurity using predictive models like LSTM, TCN, transformers, and a hybrid TCN+LSTM, evaluated in SISO and MIMO configurations. Additionally, anomaly detection techniques such as Isolation Forest, LOF, Elliptic Envelope, and One-Class SVM were assessed with metrics like AUC, F1-Score, and confusion matrices under varying noise and contamination levels. Results highlight LSTM as the most robust model for temporal patterns, Elliptic Envelope as excelling in anomaly detection in original data, and transformers as effective with augmented data, enhancing system resilience against cyber threats.

Keywords: SISO, MIMO, FDIA, LSTM, TCN, Transformers, Elliptic Envelope, Anomalies.

Índice general

Índice general	8
Índice de figuras	11
1. Introducción	13
1.1. Antecedentes	13
1.2. Definición del Problema	14
1.3. Objetivos	14
1.3.1. Objetivo General	14
1.3.2. Objetivos Específicos	14
1.4. Justificación	15
2. Estado del Arte	16
2.1. Ataques de Inyección de Datos Falsos (FDIA)	16
2.2. Modelos Predictivos para Datos de Demanda Energética	16
2.3. Algoritmos de Detección de Anomalías	17

2.4. Análisis Espectral y Densidad Espectral de Potencia	18
3. Metodología	19
3.1. Propuesta	19
3.2. Análisis de Fourier y Densidad Espectral de Potencia	21
3.3. Arquitectura de los Modelos	24
3.3.1. LSTM (Long Short-Term Memory)	24
3.3.2. TCN (Temporal Convolutional Network)	24
3.3.3. Modelo Híbrido: TCN + LSTM	25
3.3.4. Transformers	25
3.4. Implementación MIMO (Multiple Input, Multiple Output)	26
3.5. Modelos Predictivos para Generación de Datos Sintéticos	27
3.5.1. Comparación de Datos Originales y Sintéticos	27
3.5.2. Detección de Anomalías	28
4. Resultados	30
4.1. Evaluación de Modelos Predictivos	30
4.1.1. Comparación Inicial: LSTM, TCN y Transformers	30
4.1.2. Comparación Posterior: LSTM vs Modelos Híbridos	32
4.2. Generación de Datos Sintéticos	33
4.3. Evaluación de la Contaminación del Dataset Aumentado	34
4.4. Detección de Anomalías con Factores de Ruido	36

	10
4.5. Detección de Anomalías: Algoritmos	36
4.5.1. Comparación de Algoritmos	40
Conclusiones	43
Bibliografía	45
Anexos	47

Índice de figuras

3.1. Vista inicial del dataset de Demanda	20
3.2. Señal Original en el Dominio Temporal para el Transformador 10023	21
3.3. Transformada de Fourier (FFT) en el Dominio de Frecuencias para el Transformador 10023	21
3.4. Dominio de Frecuencias	22
3.5. Densidad Espectral de Potencia (PSD) del Transformador 10023, donde se destacan las frecuencias dominantes.	22
4.1. Promedio de métricas de evaluación (RMSE y R^2) para LSTM, TCN y Transformers.	31
4.2. Promedio de métricas de evaluación (RMSE y R^2) para LSTM y modelos híbridos.	33
4.3. Resumen de la generación de datos sintéticos utilizando el modelo LSTM MIMO.	34
4.4. Comparación de las distribuciones del dataset original vs aumentado.	34
4.5. Boxplots para visualización de Distribución de Datos con Anomalías y Diferentes Factores de Ruido (2.0 a 8.0)	36
4.6. Transformada de Fourier (FFT) en el Dominio de Frecuencias para el Transformador 30118	47

- 4.7. Densidad Espectral de Potencia (PSD) del Transformador 30118,
donde se destacan las frecuencias dominantes. 48
- 4.8. Transformada de Fourier (FFT) en el Dominio de Frecuencias para
el Transformador 10729 48
- 4.9. Densidad Espectral de Potencia (PSD) del Transformador 10729,
donde se destacan las frecuencias dominantes. 49

Capítulo 1

Introducción

1.1. Antecedentes

El paso hacia fuentes de energía renovable ha crecido exponencialmente debido a la preocupación mundial por el cambio climático y su efecto en energías obsoletas ya establecidas. Sin embargo, estos nuevos sistemas energéticos, como lo son los paneles solares, dependen cada vez más de tecnologías avanzadas y el uso intensivo de sistemas digitales, lo que los hace propensos a ataques cibernéticos. Es así que la ciberseguridad en los sistemas energéticos ha emergido como una prioridad crítica, teniendo en la mira incidentes relacionados con manipulaciones maliciosas de datos y ataques a la infraestructura energética [17, 6].

Estas nuevas estrategias de sistemas energéticos requieren del uso de tecnologías modernas tanto para su información como para su comunicación, haciendo referencia al control y gestión de los mismos. Dicha dependencia genera falencias considerables, pues los vuelven susceptibles a eventos como la inyección de datos falsos (False Data Injection Attacks, FDIA, por sus siglas en inglés), que pueden causar desequilibrios en la oferta y demanda de energía, afectando así la eficiencia y seguridad del sistema [17].

En este proyecto, como en aquellos de la literatura, el aprendizaje automático se considera una herramienta para conseguir una solución efectiva al abordar estas amenazas, permitiendo la detección y mitigación de ataques mediante el análisis de patrones en los datos de demanda de energía. Al hacer uso de herramientas como las redes neuronales recurrentes (RNN), redes convolucionales temporales

(TCN) y transformers, es posible desarrollar modelos predictivos que establezcan como estándar un comportamiento normal de dichos sistemas [15, 8, 16].

1.2. Definición del Problema

El problema que se aborda en este proyecto es la vulnerabilidad de los sistemas energéticos y su digitalización ante ataques compuestos por la manipulación de datos, específicamente los ataques de inyección de datos falsos (FDIA); situaciones que pueden alterar los patrones de generación y demanda de energía reportados, afectando la eficiencia del sistema energético establecido y también la confiabilidad del mismo [6].

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar modelos predictivos para detectar anomalías en datos de demanda energética.

1.3.2. Objetivos Específicos

1. Identificar patrones y periodicidades.
2. Diseñar y escoger un modelo predictivo.
 - Evaluar el rendimiento de los modelos con métricas respectivas.
 - Generación de datos sintéticos/predichos para aumentarlos al dataset original.
3. Entrenar algoritmos de detección de anomalías.
 - Usar las métricas respectivas para su evaluación y comparación.
4. Seleccionar el modelo más efectivo para la detección de anomalías.

1.4. Justificación

Se pretende que el desarrollo de modelos de aprendizaje y predicción sirva de ayuda para detectar anomalías y comportamientos sospechosos en fases tempranas del funcionamiento real del sistema energético analizado. Debido a que el sistema cuenta con una gran cantidad de datos provenientes de numerosos transformadores involucrados, y que además se está tratando con series temporales, se consideró que el uso exclusivo de métodos matemáticos o estadísticos limitaría el alcance del proyecto.

Es por esta razón que se usa el aprendizaje automático como enfoque principal. Esperando de esta manera la posibilidad mayor de identificar patrones normales y detectar de manera eficiente patrones anómalos en grandes volúmenes de datos futuros, convirtiéndose en una solución adecuada para enfrentar amenazas cibernéticas [17, 15, 16].

Capítulo 2

Estado del Arte

2.1. Ataques de Inyección de Datos Falsos (FDIA)

Las FDIA representan una amenaza para los sistemas energéticos modernos, pues buscan alterar la información de generación y demanda de energía, afectando la estabilidad y confiabilidad del sistema. Según Zhang et al. (2020), estos ataques pueden ser detectados y mitigados al hacer uso de diferentes estrategias, entre ellas las redes neuronales recurrentes (RNN) y convolucionales (CNN), por su capacidad para modelar datos de series temporales y patrones complejos inherentes al sistema energético [17].

2.2. Modelos Predictivos para Datos de Demanda Energética

Entre las técnicas más avanzadas se encuentran:

- **Redes Neuronales Recurrentes (RNN):** particularmente su variante Long Short-Term Memory (LSTM), son específicamente utilizadas en series temporales por su capacidad para manejar dependencias a largo plazo. Estas redes han demostrado ser efectivas para modelar patrones en datos de demanda energética, como lo discuten Staudemeyer y Morris (2019) [13].

- **Redes Convolucionales Temporales (TCN):** ofrecen una alternativa eficiente a las LSTM pues utilizan convoluciones dilatadas para capturar dependencias temporales de largo alcance. Según O’Shea y Nash (2015), las TCN permiten un procesamiento paralelo que mejora la eficiencia computacional, volviéndose óptimas para sistemas con una gran cantidad de datos [8].
- **Transformers:** como lo señalan Wang et al. (2022), son útiles en el análisis de series temporales pues hacen uso de mecanismos de autoatención para modelar relaciones complejas entre diferentes puntos temporales. Su aplicación en datos energéticos ha mostrado resultados buenos [16].
- **Modelos Híbridos (LSTM + TCN):** al combinar las fortalezas de LSTM y TCN, los modelos híbridos son capaces de manejar relaciones temporales complejas mientras aprovechan la eficiencia computacional de las convoluciones dilatadas.

2.3. Algoritmos de Detección de Anomalías

Para complementar los modelos predictivos, se han implementado algoritmos especializados en la detección de anomalías que destacan por su capacidad de identificar desviaciones en grandes volúmenes de datos:

- **Isolation Forest:** Este algoritmo, introducido por Liu et al. (2008), utiliza la construcción de árboles de aislamiento para identificar puntos anómalos al calcular su "profundidad de aislamiento". Es eficiente para detectar anomalías en datasets grandes y de alta dimensionalidad, y ha sido ampliamente aplicado en ciberseguridad y detección de anomalías en sistemas energéticos [7].
- **One-Class SVM:** como se lee en lo propuesto por Schölkopf et al. (2001), este algoritmo aprende un hiperplano que separa los datos normales de las anomalías en el espacio de características. Su flexibilidad y adaptabilidad lo hacen ideal para aplicaciones en entornos dinámicos [?].
- **Local Outlier Factor (LOF):** propuesto por Breunig et al. (2000), se basa en la densidad local de los puntos, clasificando como anomalías aquellos que

se encuentran en regiones de menor densidad en comparación con sus vecinos. Este enfoque es particularmente útil en datasets complejos con patrones densos [3].

- **Elliptic Envelope:** Este método, basado en el trabajo de Rousseeuw y Driessen (1999), asume una distribución gaussiana multivariada para los datos normales y detecta anomalías como puntos fuera de la elipse definida por esta distribución. Es una opción simple y efectiva para datasets con características bien definidas [9].

2.4. Análisis Espectral y Densidad Espectral de Potencia

El análisis espectral es una técnica fundamental para identificar patrones cíclicos en datos de series temporales, especialmente en sistemas energéticos. Utilizando herramientas como la Transformada de Fourier y la Densidad Espectral de Potencia (PSD), es posible determinar la frecuencia fundamental y los armónicos dominantes en los datos.

En este proyecto, el análisis espectral confirmó un ciclo de 24 horas en los datos de demanda energética, lo que respalda el uso de ventanas temporales equivalentes para los modelos predictivos. Según Staudemeyer y Morris (2019), este enfoque mejora la precisión de los modelos al garantizar que capturen adecuadamente la periodicidad de los datos [13].

Capítulo 3

Metodología

3.1. Propuesta

La propuesta específica incluye el desarrollo y evaluación de varios modelos: LSTM, TCN, un modelo híbrido TCN+LSTM, y Transformers, cada uno implementado tanto en configuraciones SISO (Single Input, Single Output) con fin de encontrar el que mejor rendimiento tiene para implementarlo en configuración MIMO (Multiple Input, Multiple Output). Para ello, se utilizarán métricas como el error cuadrático medio (MSE), error absoluto medio (MAE), el RMSE y el valor R^2 , para evaluar el rendimiento de cada modelo. Además, se implementará el análisis de Fourier y la densidad espectral de potencia para explorar características fundamentales de la señal de demanda.

Cada uno de estos modelos se entrenará utilizando un conjunto de datos de 25,000 mediciones temporales por transformador (el dataset completo tiene un total de 17 transformadores). El primer bloque de datos (0 a 17,500) se utilizará para la fase de entrenamiento, el siguiente bloque (17,501 a 22,500) para la fase de validación, y el último bloque (últimos 2500 datos aproximadamente) para la fase de testing/pruebas. Se nota que cada medición corresponde a una hora en la serie temporal del dataset, por lo que las 25000 mediciones temporales corresponden a 25000 horas.

Para gestionar estas grandes cantidades de datos y entrenar los modelos mencionados de manera eficiente, se utilizan herramientas como el **DataLoader** y el **DataModule** de PyTorch Lightning. Estas herramientas permiten una carga y

procesamiento concurrente, facilitando la administración de conjuntos de datos grandes y complejos. Adicionalmente, PyTorch Lightning facilita la estructuración de los modelos a definirse, permitiendo un desarrollo más modular y organizado [4].

Además, se propone utilizar la plataforma *Weights and Biases (Wandb)* para realizar un seguimiento detallado de los experimentos de aprendizaje automático. *Wandb* permite visualizar métricas en tiempo real, comparar diferentes modelos y realizar un seguimiento detallado del rendimiento del modelo, facilitando así el reconocimiento de problemáticas como el *overfitting* y *underfitting*, para de esta manera modificar los modelos y seleccionar el mejor para el objetivo de detección de FDIAs [2].

Como parte del proceso de aprendizaje, se utiliza el dataset proporcionado de demanda energética (*Demand.csv*). Se considera que los IDs, únicos para cada transformador se encuentran en la primera columna del dataset. A continuación se muestra una vista inicial en la figura 3.1:

Demand									
	10729	11031	11081	11132	11139	11194	11607	30051	30079
1									
2	69.04000000000002	39.41917999999998	14.39	37.43	71.99999999999999	63.570000000000014	43.430000000000001	25.26	21.24
3	71.92	38.38917999999999	13.999999999999998	38.940000000000026	70.99	62.220000000000056	38.57	24.630000000000003	21.1
4	67.390000000000001	32.539180000000001	13.96	62.780000000000005	63.180000000000014	60.76	34.300000000000001	26.05	21.16
5	65.759999999999999	29.629179999999995	13.909999999999998	31.249999999999975	50.569999999999998	49.189999999999999	31.409999999999997	31.49	21.03
6	64.36	27.999180000000001	14.010000000000002	30.219999999999985	47.269999999999975	41.410000000000001	26.140000000000008	34.2	19.62
7	63.38	26.959179999999986	14.09	31.799999999999976	42.96	38.509999999999984	25.110000000000007	37.91	15.31
8	64.95	27.879180000000001	13.99	28.619999999999965	44.490000000000016	40.680000000000014	24.62	42.82	15.239999999999998
9	65.42	28.169179999999987	14.71	31.559999999999998	46.189999999999998	46.670000000000003	29.159999999999997	48.86	15.27
10	68.820000000000001	29.779180000000007	16.599999999999998	36.84	54.439999999999998	50.960000000000005	31.15	48.929999999999999	15.44
11	74.84	34.63918	15.82	43.780000000000003	69.6	75.020000000000001	34.68	47.18	15.3
12	71.710000000000001	46.359180000000001	15.47	47.260000000000003	77.929999999999998	82.610000000000001	40.299999999999999	46.62	15.22
13	73.100000000000002	44.309180000000002	15.739999999999998	51.140000000000015	88.679999999999998	86.47	48.579999999999999	46.61	15.4
14	74.06	42.069180000000002	15.829999999999998	85.01	99.71	75.530000000000006	42.73	46.26	15.47
15	73.0	41.159180000000006	14.29	93.540000000000003	97.249999999999999	89.839999999999997	40.64	46.38	15.54
16	75.489999999999998	44.049180000000001	14.34	78.399999999999998	98.220000000000003	86.609999999999997	44.570000000000014	47.16	15.509999999999998
17	80.21	43.31918	14.929999999999998	55.430000000000001	102.889999999999992	91.510000000000003	50.419999999999995	54.03	15.48
18	81.430000000000002	58.939180000000004	15.05	71.939999999999997	124.38	123.710000000000002	68.860000000000003	51.8	15.509999999999998
19	83.95	68.579180000000002	14.959999999999996	92.000000000000003	142.649999999999998	148.249999999999997	74.98	42.74	15.4
20	82.96	67.25918	15.249999999999998	96.340000000000003	128.410000000000005	131.220000000000003	64.839999999999997	28.42	15.46
21	76.73	58.659179999999999	14.11	77.979999999999992	116.249999999999996	109.079999999999998	50.180000000000014	27.75	15.87
22	76.45	59.149180000000003	14.71	62.440000000000004	98.9	98.89	48.640000000000003	27.74	15.89

Figura 3.1: Vista inicial del dataset de Demanda

3.2. Análisis de Fourier y Densidad Espectral de Potencia

Se implementó código Matlab con el que se obtuvo la transformada de Fourier que convierte la señal, que originalmente se encontraba en el dominio temporal mostrado en la figura 3.2, al dominio de frecuencia mostrado en la figura 3.3:

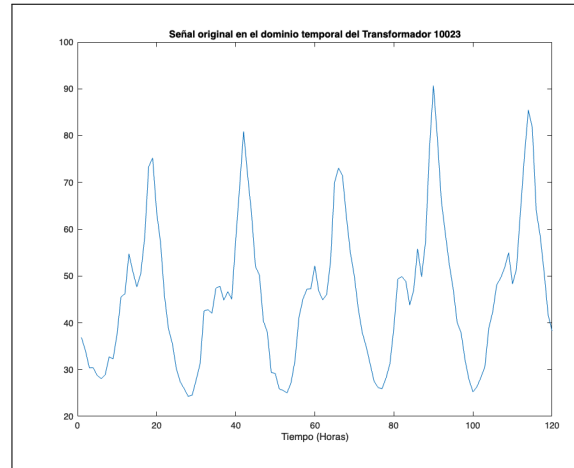


Figura 3.2: Señal Original en el Dominio Temporal para el Transformador 10023

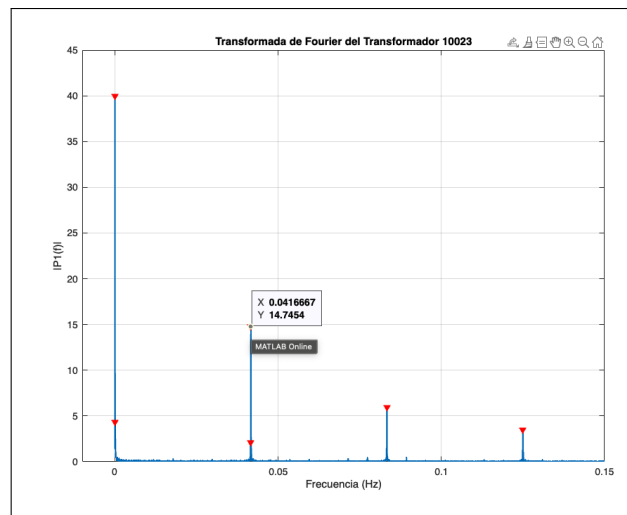


Figura 3.3: Transformada de Fourier (FFT) en el Dominio de Frecuencias para el Transformador 10023

Dominio de Frecuencias (FFT de la Señal Original)

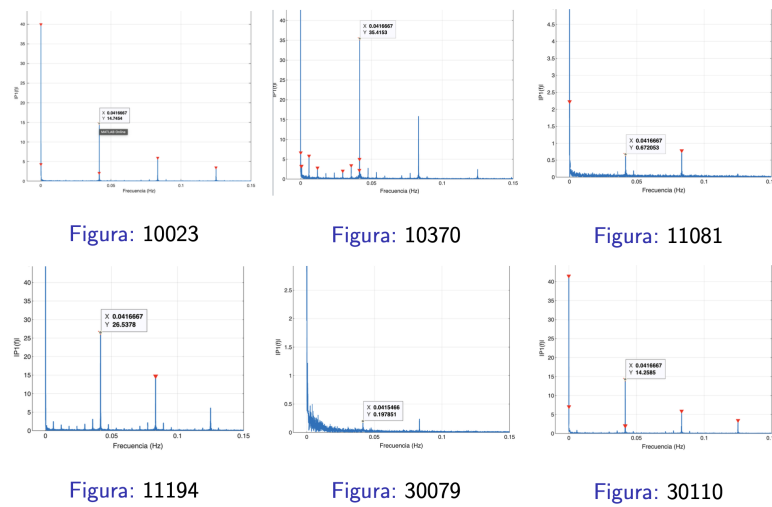


Figura 3.4: Dominio de Frecuencias

Una vez en este dominio, es posible identificar las frecuencias dominantes en las señales de demanda y generación al analizar los primeros picos, los cuales representan los armónicos de la señal. Este análisis permitió determinar la frecuencia fundamental y los armónicos presentes en los datos. El código desarrollado en Matlab realiza la transformada de Fourier sobre las señales de demanda y generación y calcula la densidad espectral de potencia (PSD) utilizando el método de Welch, la cual se puede observar en la figura 3.5:

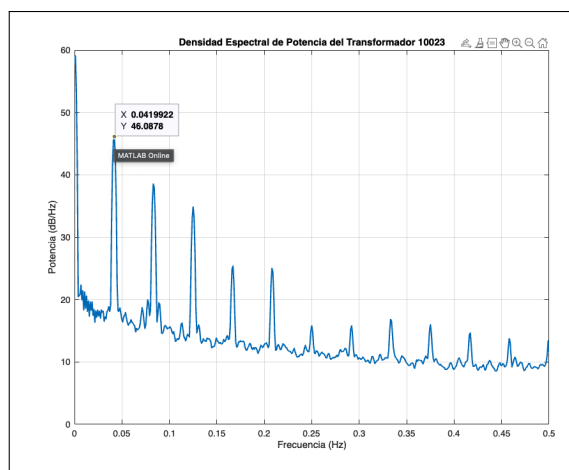


Figura 3.5: Densidad Espectral de Potencia (PSD) del Transformador 10023, donde se destacan las frecuencias dominantes.

Los primeros picos encontrados en ambas gráficas representan las frecuencias principales que componen la señal. En la sección de Anexos, en las figuras 4.6, 4.7, 4.8 y 4.9, se puede observar el mismo comportamiento en cuanto a frecuencias principales para otros transformadores, mostrando un comportamiento general y no individual.

Es así que la frecuencia fundamental se identificó como aproximadamente **0.0417 [Hz]**, lo cual implica un ciclo completo cada **24 horas**. Este valor se obtiene usando la relación 3.1:

$$T = \frac{1}{f} \quad (3.1)$$

donde T es el periodo en horas y f es la frecuencia en Hz (considerando que la frecuencia de muestreo está en horas). Para $f \approx 0,0417$ Hz:

$$T = \frac{1}{0,0417} \approx 24 \text{ horas} \quad (3.2)$$

lo que sugiere que la señal tiene un comportamiento cíclico cada 24 horas, consistente con patrones típicos en datos de demanda energética que varían según el ciclo diario.

De esta forma es posible establecer una relación con el tamaño de ventana a utilizar en el entrenamiento de modelos predictivos. Es importante que el tamaño de la ventana de entrada represente un ciclo completo de la señal para poder modelar adecuadamente las dependencias temporales. Debido a que la frecuencia fundamental tiene un periodo de 24 horas, un tamaño de ventana del mismo valor sería ideal para capturar toda la variabilidad cíclica diaria tanto de la demanda como de la generación energética, pues el modelo puede aprender patrones repetitivos y prever anomalías de manera más eficiente.

3.3. Arquitectura de los Modelos

3.3.1. LSTM (Long Short-Term Memory)

El modelo LSTM utiliza una ventana deslizante con datos históricos (por ejemplo, 24 horas) como entrada, procesando estas ventanas para extraer patrones temporales y predecir el valor de la próxima hora. La arquitectura fue configurada con un número variable de capas ocultas y celdas LSTM.

Durante el entrenamiento, se probaron diferentes tamaños de ventana (`range(1, 25)`) para determinar experimentalmente el tamaño óptimo. Además, se registraron las métricas `train loss` y `val loss` para monitorear el desempeño del modelo y ajustar los hiperparámetros. La normalización de los datos se realizó escalando los valores entre 0 y 1 para mejorar la convergencia del modelo.

En su configuración final, el modelo incluyó:

- 128 unidades ocultas.
- 3 capas LSTM.
- Una tasa de `dropout` del 30 %.

La implementación fue realizada utilizando PyTorch Lightning, facilitando la estructuración y gestión eficiente del modelo.

3.3.2. TCN (Temporal Convolutional Network)

El modelo TCN se desarrolló para procesar ventanas deslizantes de datos históricos, similar al enfoque utilizado con LSTM. Las TCN, a diferencia de las RNN, procesan los datos de manera paralela utilizando convoluciones dilatadas, lo que las hace más eficientes para tareas con grandes volúmenes de datos.

Este modelo también exploró diferentes tamaños de ventana (`range(1, 25)`) para encontrar la configuración óptima, con un registro constante de las métricas `train loss` y `val loss`. Los datos fueron normalizados entre 0 y 1 para mejorar el aprendizaje.

La arquitectura final del modelo incluyó:

- 64 unidades ocultas.
- 3 capas convolucionales.
- Dilatación exponencial (1, 2, 4).
- Una tasa de `dropout` del 20 %.

La implementación fue realizada en PyTorch Lightning, optimizando el manejo de las dependencias de largo alcance en los datos.

3.3.3. Modelo Híbrido: TCN + LSTM

El modelo híbrido combina las fortalezas de las TCN y las LSTM, procesando datos históricos a través de convoluciones dilatadas para capturar patrones temporales y utilizando celdas LSTM para modelar dependencias secuenciales complejas.

El enfoque experimental incluyó el ajuste de los tamaños de ventana y el monitoreo de `train loss` y `val loss`, destacando las siguientes configuraciones:

- 128 unidades ocultas.
- 3 capas convolucionales (TCN) seguidas por 2 capas LSTM.
- Dilatación exponencial (1, 2, 4) en la parte convolucional.
- Una tasa de `dropout` del 25 %.

Este modelo fue entrenado para predecir valores simultáneamente en múltiples series temporales.

3.3.4. Transformers

La arquitectura del transformer se implementó para procesar secuencias de datos históricos mediante capas de autoatención, identificando las características

más relevantes en cada paso temporal. Aunque los transformers no procesan datos temporales de manera directa, utilizan codificación posicional para incorporar el orden temporal en los datos.

Las configuraciones experimentales incluyeron:

- 64 unidades ocultas.
- 2 capas de autoatención con 4 cabezas cada una.
- Una tasa de **dropout** del 10 %.

El modelo se entrenó para predecir el valor de la próxima hora y optimizar el tamaño de la ventana deslizante.

3.4. Implementación MIMO (Multiple Input, Multiple Output)

El enfoque MIMO extendió el modelo LSTM para procesar múltiples columnas del dataset simultáneamente. Cada entrada representaba las mediciones de varios transformadores durante un rango de tiempo, mientras que la salida consistía en predicciones simultáneas para todas las columnas.

Este enfoque aprovechó la arquitectura previamente definida y fue configurado para:

- Predecir múltiples salidas con base en una ventana de entrada común.
- Mantener una tasa de **dropout** del 30 % para evitar sobreajuste.
- Escalar el entrenamiento para diferentes configuraciones de datos mediante PyTorch Lightning.

3.5. Modelos Predictivos para Generación de Datos Sintéticos

Se exploraron diferentes arquitecturas de modelos predictivos para la generación de datos sintéticos. Entre estas arquitecturas, se incluyeron LSTM, TCN y Transformers. Después de los resultados, que se pueden observar en la siguiente sección de este trabajo, se determinó que el modelo LSTM presentaba el mejor rendimiento en métricas como R^2 y RMSE, por lo que se seleccionó para implementar un esquema MIMO (*Multiple Input, Multiple Output*) y generar datos sintéticos.

Configuración del Modelo LSTM

El modelo LSTM utilizado para generación de datos fue configurado con las siguientes características:

- **Entrada:** Ventana deslizante de 24 horas con datos históricos de demanda energética.
- **Arquitectura:** 3 capas LSTM con 128 unidades ocultas, dropout del 30%.
- **Salida:** Predicción del valor de la próxima hora.
- **Normalización:** Escalado entre 0 y 1 para mejorar el aprendizaje.

Generación de Datos Sintéticos con MIMO

Se utilizó el modelo LSTM entrenado para generar 500 datos sintéticos adicionales, basados en ventanas deslizantes de datos históricos (últimas 24 horas). El conjunto total de datos resultante consistió en 25,484 observaciones (24,984 originales + 500 sintéticos).

3.5.1. Comparación de Datos Originales y Sintéticos

Para garantizar la validez de los datos generados y verificar que no se introdujeran contaminaciones, se compararon las distribuciones estadísticas de los datos

originales y los datos aumentados mediante las siguientes técnicas:

- **Prueba de Kolmogorov-Smirnov (KS):** Evaluación de la similitud entre las distribuciones.
- **Elliptic Envelope:** Algoritmo de detección de anomalías utilizado para identificar puntos fuera de la distribución gaussiana.

Se generaron visualizaciones de las distribuciones para comparar los datos originales y sintéticos bajo diferentes transformadores.

3.5.2. Detección de Anomalías

Una vez validados los datos generados, se procedió a la detección de anomalías tanto en el conjunto original como en el conjunto aumentado. Los métodos empleados incluyeron Isolation Forest, One-Class SVM, Local Outlier Factor (LOF) y Elliptic Envelope.

Preprocesamiento de los Datos

1. Se dividieron los datos en conjuntos de entrenamiento (80 %) y prueba (20 %), manteniendo el orden temporal.
2. Se introdujeron anomalías en el conjunto de prueba añadiendo ruido proporcional a la desviación estándar de los datos, controlado por un *noise factor*. Las anomalías se introdujeron en el 5 % de las observaciones seleccionadas aleatoriamente.

Entrenamiento y Evaluación de los Modelos

Cada algoritmo fue entrenado de forma focalizada para cada uno de los 17 transformadores descritos en el dataset. Las configuraciones probadas incluyeron múltiples niveles de contaminación (*contamination values*) y factores de ruido. Las métricas evaluadas incluyeron:

- **Precision-Recall AUC (PR-AUC):** Para medir el balance entre precisión y recall.
- **Matriz de Confusión:** Para analizar los falsos positivos (FP) y los falsos negativos (FN).
- **ROC-AUC:** Para evaluar la capacidad del modelo de separar anomalías de datos normales.
- **F1-Score:** como la métrica de mayor interés por su relación con los valores de precisión y recall.

Visualización y Análisis de Resultados

Se generaron boxplots para visualizar la distribución de los datos bajo diferentes escenarios, incluyendo niveles de contaminación y factores de ruido, antes y después de haber añadido las anomalías mencionadas. Esto permitió identificar visualmente los efectos de las anomalías y validar la robustez de los modelos implementados.

Capítulo 4

Resultados

4.1. Evaluación de Modelos Predictivos

En este apartado se presentan los resultados de la evaluación de los modelos predictivos desarrollados para la tarea de predicción de series temporales en transformadores eléctricos. Se utilizó una ventana de 24 horas y las métricas principales fueron el *Mean Squared Error* (MSE), el *Root Mean Squared Error* (RMSE), el error absoluto medio (*Mean Absolute Error*, MAE) y el coeficiente de determinación (R^2).

El análisis se dividió en dos etapas:

1. Comparación inicial entre los modelos **LSTM**, **TCN** y **Transformers**.
2. Comparación posterior entre el modelo LSTM y los modelos híbridos (combinación de TCN y LSTM).

4.1.1. Comparación Inicial: LSTM, TCN y Transformers

En la Tabla 4.1.1 se muestran los resultados obtenidos para cada transformador con las tres arquitecturas evaluadas. Los modelos fueron evaluados en el conjunto de prueba, y las métricas promedio para cada modelo se presentan en la Figura 4.1.1.

En la Tabla 4.1.1 se muestran los resultados obtenidos para cada transformador con las tres arquitecturas evaluadas. Los modelos fueron evaluados en el conjunto de prueba, y las métricas promedio para cada modelo se presentan en la Figura 4.1.1.

M./T.	10023	10193	10292	10370	10729	11031	11081	11132	11139	11194	11607	30051	30079	30086	30095	30110	30118
	LSTM (improved: 128 hidden dim., 3 layers, dropout 30%)																
MSE	0.0048	0.0059	0.0091	0.0045	0.0035	0.0058	0.0046	0.0096	0.0047	0.0035	0.0091	0.0025	0.0013	0.0070	0.0016	0.0055	0.0030
MAE	0.0507	0.0556	0.0720	0.0528	0.0430	0.0589	0.0330	0.0753	0.0526	0.0438	0.0725	0.0354	0.0207	0.0539	0.0279	0.0532	0.0360
RMSE	0.0691	0.0771	0.0956	0.0670	0.0589	0.0762	0.0675	0.0979	0.0685	0.0590	0.0956	0.0500	0.0356	0.0835	0.0397	0.0743	0.0543
R²	0.8807	0.7732	0.6468	0.9199	0.8315	0.8070	0.7660	0.7712	0.8216	0.8641	0.6539	0.9617	0.9804	0.8941	0.9819	0.8175	0.9402
	TRANSFORMERS																
MSE	0.0093	0.0080	0.0100	0.0089	0.0048	0.0087	0.0068	0.0173	0.0067	0.0064	0.0100	0.0070	0.0014	0.0121	0.0084	0.0086	0.0094
MAE	0.0745	0.0679	0.0780	0.0755	0.0541	0.0723	0.0542	0.0996	0.0601	0.0607	0.0763	0.0552	0.0195	0.0712	0.0563	0.0713	0.0792
RMSE	0.0964	0.0893	0.1002	0.0945	0.0690	0.0934	0.0824	0.1317	0.0820	0.0797	0.1002	0.0838	0.0375	0.1102	0.0919	0.0926	0.0971
R²	0.7676	0.6958	0.6123	0.8404	0.7690	0.7104	0.6518	0.5859	0.7447	0.7521	0.6195	0.8925	0.9783	0.8157	0.9032	0.7161	0.8091
	TCN																
MSE	0.0091	0.0081	0.0100	0.0096	0.0043	0.0080	0.0069	0.0158	0.0071	0.0068	0.0098	0.0079	0.0021	0.0109	0.0081	0.0079	0.0059
MAE	0.0755	0.0700	0.0762	0.0776	0.0498	0.0691	0.0580	0.0963	0.0606	0.0596	0.0734	0.0668	0.0363	0.0663	0.0596	0.0660	0.0483
RMSE	0.0954	0.0902	0.1001	0.0980	0.0658	0.0897	0.0832	0.1257	0.0841	0.0822	0.0989	0.0886	0.0457	0.1046	0.0901	0.0889	0.0770
R²	0.7723	0.6896	0.6133	0.8285	0.7898	0.7331	0.6450	0.6231	0.7316	0.7364	0.6294	0.8799	0.9677	0.8338	0.9070	0.7384	0.8799

Cuadro 4.1: Resultados de evaluación por transformador para LSTM, TCN y Transformers (Ventana de 24 horas).

```

RMSE Promedio – LSTM: 0.0688
RMSE Promedio – Transformers: 0.0901
RMSE Promedio – TCN: 0.0887

-----

R2 Promedio – LSTM: 0.8419
R2 Promedio – Transformers: 0.7567
R2 Promedio – TCN: 0.7646

```

Figura 4.1: Promedio de métricas de evaluación (RMSE y R^2) para LSTM, TCN y Transformers.

Resultados Observados

- **LSTM** presentó el mejor rendimiento global, con un **RMSE promedio de 0.0688** y un **R^2 promedio de 0.8419**.
- **TCN** mostró un rendimiento intermedio, con un **RMSE promedio de 0.0887** y un **R^2 promedio de 0.7646**.

- **Transformers** tuvo el menor rendimiento en términos de predicción, con un RMSE promedio de **0.0901** y un R^2 promedio de **0.7567**.

Los resultados confirman que el modelo LSTM, con una arquitectura de **128 hidden units, 3 capas y 30 % de dropout**, es más adecuado para capturar las dependencias temporales y los patrones cíclicos en los datos de transformadores eléctricos.

4.1.2. Comparación Posterior: LSTM vs Modelos Híbridos

Dado el buen desempeño del modelo LSTM, se exploró la posibilidad de mejorar los resultados mediante modelos híbridos que combinan **TCN** y **LSTM**. Para esto, se evaluaron dos configuraciones:

1. **Híbrido estándar:** Una arquitectura básica de 64 *hidden units*, 1 capa y 20 % de *dropout*.
2. **Híbrido mejorado:** Una arquitectura optimizada con 128 *hidden units*, 3 capas y 30 % de *dropout*.

Los resultados detallados para cada transformador se presentan en la Tabla 4.1.2. Los promedios de las métricas RMSE y R^2 se ilustran en la Figura 4.1.2.

M./T.	10023	10193	10292	10370	10729	11031	11081	11132	11139	11194	11607	30051	30079	30086	30095	30110	30118
	HYBRID (TCN + LSTM (standard: 64 hidden dim., 1 layer, dropout 20 %))																
MSE	0.0047	0.0064	0.0089	0.0046	0.0040	0.0060	0.0072	0.0091	0.0050	0.0032	0.0103	0.0023	0.0031	0.0064	0.0015	0.0050	0.0030
MAE	0.0498	0.0610	0.0711	0.0523	0.0461	0.0581	0.0584	0.0734	0.0536	0.0417	0.0759	0.0353	0.0405	0.0496	0.0275	0.0522	0.0366
RMSE	0.0687	0.0801	0.0946	0.0677	0.0632	0.0775	0.0849	0.0955	0.0705	0.0568	0.1017	0.0484	0.0554	0.0797	0.0392	0.0707	0.0546
R²	0.8820	0.7552	0.6547	0.9181	0.8064	0.8008	0.6299	0.7825	0.8111	0.8743	0.6082	0.9642	0.9525	0.9035	0.9823	0.8348	0.9397
	HYBRID (TCN + LSTM (improved: 128 hidden dim., 3 layers, dropout 30 %))																
MSE	0.0051	0.0060	0.0094	0.0059	0.0038	0.0060	0.0091	0.0085	0.0054	0.0040	0.0099	0.0023	0.0025	0.0066	0.0027	0.0046	0.0030
MAE	0.0521	0.0589	0.0732	0.0596	0.0475	0.0593	0.0736	0.0707	0.0550	0.0458	0.0762	0.0354	0.0312	0.0531	0.0366	0.0506	0.0355
RMSE	0.0714	0.0776	0.0968	0.0769	0.0619	0.0773	0.0954	0.0924	0.0734	0.0631	0.0995	0.0484	0.0501	0.0811	0.0516	0.0681	0.0548
R²	0.8723	0.7698	0.6382	0.8943	0.8144	0.8015	0.5331	0.7960	0.7952	0.8447	0.6250	0.9641	0.9611	0.9001	0.9694	0.8465	0.9391

Cuadro 4.2: Resultados de evaluación por transformador para LSTM y modelos híbridos (TCN + LSTM).


```
RMSE Promedio - LSTM: 0.0688
RMSE Promedio - Hybrid ST: 0.0711
RMSE Promedio - Hybrid Improved: 0.0729
-----
R2 Promedio - LSTM: 0.8419
R2 Promedio - Hybrid ST: 0.8294
R2 Promedio - Hybrid Improved: 0.8215
```

Figura 4.2: Promedio de métricas de evaluación (RMSE y R^2) para LSTM y modelos híbridos.

Resultados Observados

- Aunque los modelos híbridos (en especial el **híbrido mejorado**) se acercaron al rendimiento de LSTM, **no lograron superarlo en términos globales**.
- El **híbrido mejorado** alcanzó un **RMSE promedio de 0.0729** y un **R^2 promedio de 0.8215**, mientras que el modelo LSTM mantuvo su liderazgo con un **RMSE promedio de 0.0688** y un **R^2 promedio de 0.8419**.
- El **híbrido estándar** mostró un desempeño intermedio con un **RMSE promedio de 0.0711** y un **R^2 promedio de 0.8294**.

4.2. Generación de Datos Sintéticos

Para mejorar la cantidad de datos disponibles y evaluar la robustez de los modelos predictivos, se utilizó el modelo **LSTM** en su versión *Multiple Input Multiple Output* (MIMO) previamente entrenado. Este modelo permitió generar datos sintéticos basados en ventanas de datos históricos de 24 horas, resultando en un conjunto aumentado de **25,484 registros** (24,984 datos originales + 500 datos sintéticos).

1	9	11194	11607	30051	30079	30086	30095	30110	30118	Type
2	1	63.570000000000014	43.430000000000001	25.26	21.24	7.389999999999999	29.45	61.128108	18.16	Original
3		62.2200000000000056	38.57	24.630000000000003	21.1	7.42	29.660000000000004	52.27810799999999	17.6	Original
4	4	60.76	34.300000000000001	26.05	21.16	7.919999999999999	29.629999999999995	44.898108000000015	17.770000000000003	Original
5	1	49.189999999999999	31.41	31.49	21.03	7.44	29.97	43.468108000000001	17.630000000000003	Original
6	5	41.410000000000001	26.140000000000008	34.2	19.62	7.58	29.690000000000005	41.398107999999998	17.810000000000002	Original
7		38.509999999999984	25.110000000000007	37.91	15.31	7.689999999999999	32.21	41.218108	19.16	Original
8	6	40.680000000000014	24.62	42.82	15.239999999999998	7.6099999999999985	39.370000000000005	41.518108	20.8	Original
9	1	46.670000000000003	29.16	48.86	15.27	7.73	40.420000000000001	43.278108	25.55	Original
10	1	50.960000000000005	31.15	48.92999999999999	15.44	5.66	40.250000000000014	44.828107999999999	25.08	Original
11		75.020000000000001	34.68	47.18	15.3	5.39	39.520000000000001	45.498108000000016	23.76	Original
00001		43.95	20.49	20.34	12.830000000000002	4.76	20.700000000000003	25.879999999999998	11.45	Original
32617		29.61539649963379	15.173054695129395	12.556526184082031	15.463595390319824	5.809892177581787	10.987709045410156	25.844432830810547	13.745729446411133	Synthetic
1106445		23.239072799682617	15.86574935913086	8.847222328186035	14.791451454162598	6.0974440574646	9.677447319030762	22.774065017700195	13.902968406677246	Synthetic
942383		20.81867218017578	16.32494354248047	7.95339298248291	14.690412521362305	6.299115180969238	9.883661270141602	21.472965240478516	14.206408500671387	Synthetic
032227		20.440378189086914	15.88782787322998	9.570350646972656	14.565579414367676	6.5777812004089355	10.898799896240234	20.913232803344727	14.897603034973145	Synthetic
46875		21.68755531311035	14.75487232208252	14.524840354919434	14.349946975708008	8.12528133392334	13.49801254272461	20.740375518798828	16.985580444335938	Synthetic
9082		31.01283836364746	17.99907112121582	38.82838439941406	14.402287483215332	22.553659439086914	20.986753463745117	22.285491943359375	25.787078857421875	Synthetic
21875		59.35172653198242	33.48102951049805	70.5689926147461	14.510427474975586	30.648794174194336	36.22864532470703	30.23959732055664	37.58399963378906	Synthetic
65039		59.823543548583984	33.566097259521484	96.30025482177734	14.985852241516113	29.786136627197266	77.14320373535156	32.24101638793945	59.27562713623047	Synthetic

Figura 4.3: Resumen de la generación de datos sintéticos utilizando el modelo LSTM MIMO.

4.3. Evaluación de la Contaminación del Dataset Aumentado

Se compararon las distribuciones del dataset original y del dataset aumentado para verificar la posible contaminación introducida al generar los datos sintéticos. En la Figura 4.3 se presentan histogramas que ilustran la similitud entre las distribuciones de los datos originales y los aumentados para varios transformadores. La prueba estadística de Kolmogorov-Smirnov (KS) confirmó que las distribuciones son similares ($p \geq 0,05$).

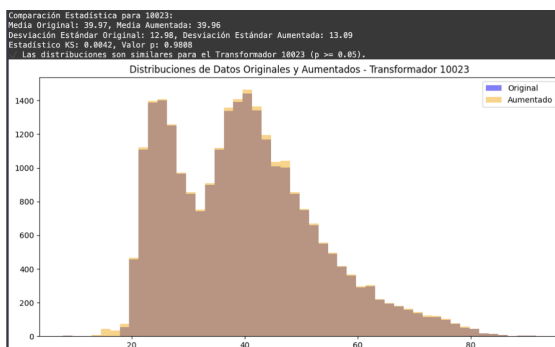
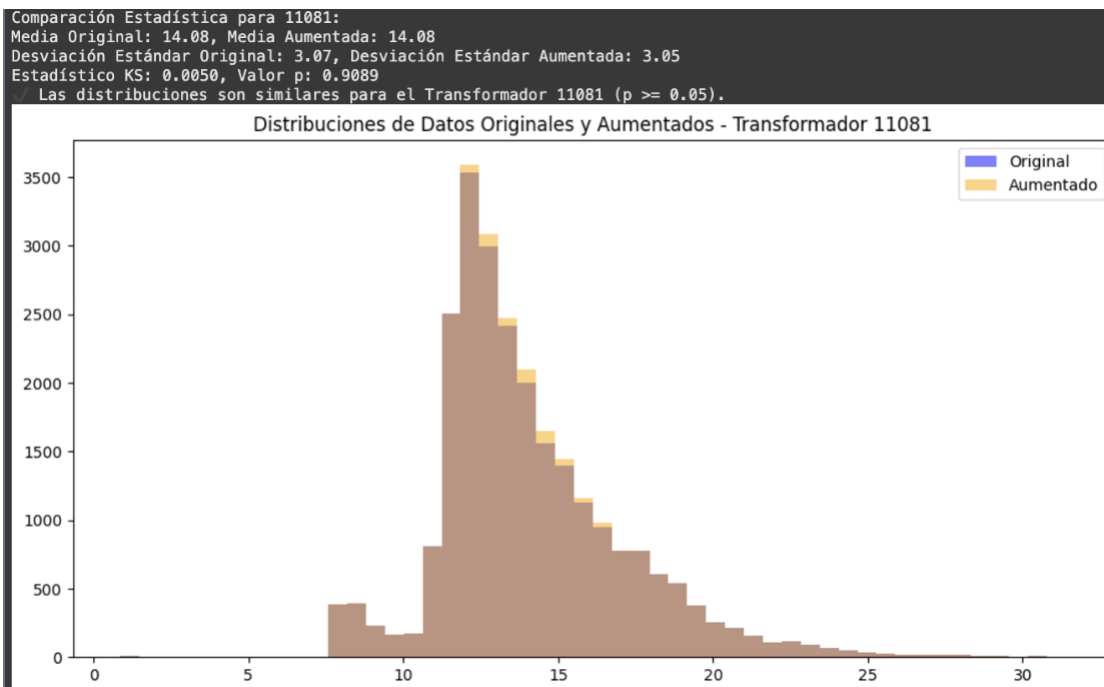
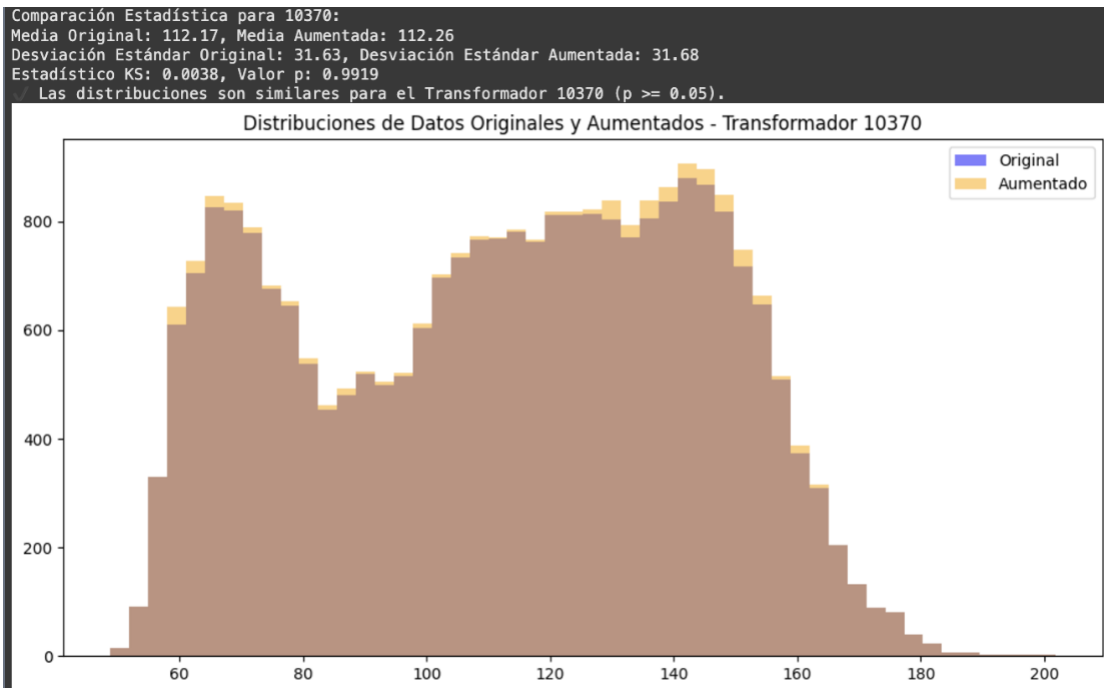


Figura 4.4: Comparación de las distribuciones del dataset original vs aumentado.



4.4. Detección de Anomalías con Factores de Ruido

Para evaluar la detección de anomalías, se generaron anomalías artificiales en los datos de prueba. Estas anomalías fueron generadas añadiendo ruido proporcional a la desviación estándar de los datos originales. En la Figura 4.5, se presentan los boxplots de un transformador específico con diferentes factores de ruido aplicados a los datos.

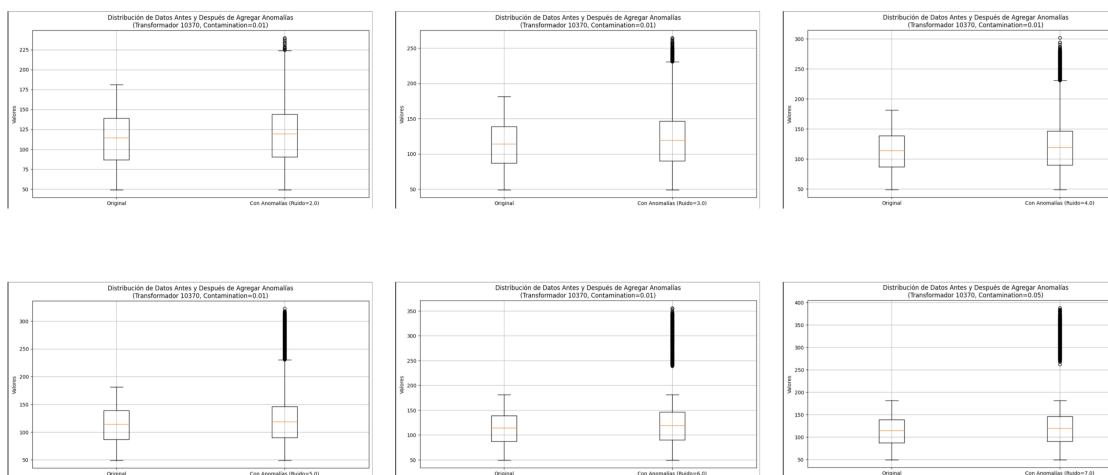


Figura 4.5: Boxplots para visualización de Distribución de Datos con Anomalías y Diferentes Factores de Ruido (2.0 a 8.0)

4.5. Detección de Anomalías: Algoritmos

Se evaluaron múltiples algoritmos para la detección de anomalías en los datasets original y aumentado. Los algoritmos evaluados incluyeron:

- Isolation Forest
- One-Class SVM
- Elliptic Envelope
- Local Outlier Factor (LOF)

En las Tabla 4.3 y 4.4, se resumen las mejores configuraciones de parámetros y los resultados obtenidos para el algoritmo **Isolation Forest** utilizando el dataset original y el dataset aumentado.

Transformador	Contamination	Noise Factor	PR-AUC	ROC-AUC	F1-Score	CM
10023	0.01	7.0	0.971698	0.996654	0.970874	[4453, 30, 0, 500]
10193	0.01	8.0	0.942478	0.992750	0.938967	[4418, 65, 0, 500]
10292	0.01	8.0	0.996112	0.998665	0.996008	[4480, 3, 1, 499]
10370	0.01	6.0	0.983559	0.998104	0.983284	[4466, 17, 0, 500]
10729	0.01	8.0	0.909836	0.987731	0.900901	[4373, 110, 0, 500]
11031	0.01	6.0	0.968838	0.993654	0.967868	[4453, 30, 3, 497]
11081	0.01	7.0	0.962107	0.995427	0.960615	[4442, 41, 0, 500]
11132	0.01	7.0	0.928082	0.990631	0.922509	[4399, 84, 0, 500]
11139	0.01	8.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]
11194	0.01	7.0	0.999002	0.999888	0.999001	[4482, 1, 0, 500]
11607	0.01	8.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]
30051	0.01	4.0	0.920875	0.989516	0.914077	[4389, 94, 0, 500]
30079	0.01	7.0	0.987329	0.998550	0.987167	[4470, 13, 0, 500]
30086	0.01	5.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]
30095	0.01	4.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]
30110	0.01	8.0	0.955373	0.994535	0.953289	[4434, 49, 0, 500]
30118	0.01	4.0	0.811721	0.966317	0.768049	[4181, 302, 0, 500]

Cuadro 4.3: Mejores Configuraciones por Transformador (Dataset Original)

Transformador	Contamination	Noise Factor	PR-AUC	ROC-AUC	F1-Score	CM
10023	0.01	6.0	0.902576	0.986839	0.892061	[4476, 121, 0, 500]
10193	0.01	8.0	0.929553	0.991081	0.924214	[4515, 82, 0, 500]
10292	0.01	7.0	0.980769	0.997825	0.980392	[4577, 20, 0, 500]
10370	0.01	4.0	0.988314	0.997804	0.988119	[4586, 11, 1, 499]
10729	0.01	6.0	0.906504	0.987492	0.896861	[4482, 115, 0, 500]
11031	0.01	7.0	0.886399	0.984011	0.871840	[4450, 147, 0, 500]
11081	0.01	7.0	0.965453	0.995084	0.964251	[4561, 36, 1, 499]
11132	0.01	6.0	0.878788	0.982597	0.862069	[4437, 160, 0, 500]
11139	0.01	7.0	0.941696	0.992821	0.938086	[4531, 66, 0, 500]
11194	0.01	7.0	0.896197	0.985752	0.884173	[4466, 131, 0, 500]
11607	0.01	8.0	0.975285	0.997172	0.974659	[4571, 26, 0, 500]
30051	0.01	7.0	0.851617	0.977050	0.825764	[4386, 211, 0, 500]
30079	0.01	7.0	0.988281	0.998695	0.988142	[4585, 12, 0, 500]
30086	0.01	5.0	1.000000	1.000000	1.000000	[4597, 0, 0, 500]
30095	0.01	5.0	1.000000	1.000000	1.000000	[4597, 0, 0, 500]
30110	0.01	7.0	0.926621	0.990646	0.920810	[4511, 86, 0, 500]
30118	0.01	4.0	0.794464	0.962040	0.741290	[4248, 349, 0, 500]

Cuadro 4.4: Mejores Configuraciones por Transformador (Dataset + 500 datos predichos)

Los resultados de la detección de anomalías utilizando **One-Class SVM** se presentan en las Tablas 4.5 y 4.6. Este algoritmo mostró un desempeño bajo, especialmente para factores de ruido bajos, comparado con el algoritmo previo.

Transformador	Nu	Noise Factor	PR-AUC	ROC-AUC	F1-Score	CM
10023	0.01	8.0	0.862319	0.978809	0.840336	[4293, 190, 0, 500]
10193	0.01	8.0	0.866746	0.953393	0.857668	[4361, 122, 33, 467]
10292	0.05	8.0	0.880518	0.982489	0.864304	[4326, 157, 0, 500]
10370	0.01	7.0	0.932526	0.991300	0.927644	[4405, 78, 0, 500]
10729	0.01	8.0	0.850509	0.975467	0.824793	[4272, 211, 1, 499]
11031	0.01	7.0	0.855619	0.977359	0.831255	[4280, 203, 0, 500]
11081	0.05	8.0	0.909836	0.987731	0.900901	[4373, 110, 0, 500]
11132	0.01	8.0	0.804878	0.964310	0.757576	[4163, 320, 0, 500]
11139	0.01	8.0	0.952080	0.994089	0.949668	[4430, 53, 0, 500]
11194	0.01	8.0	0.903226	0.986616	0.892857	[4363, 120, 0, 500]
11607	0.01	8.0	0.968969	0.995431	0.967992	[4451, 32, 1, 499]
30051	0.01	4.0	0.934028	0.991524	0.929368	[4407, 76, 0, 500]
30079	0.01	6.0	0.964326	0.992208	0.963107	[4449, 34, 4, 496]
30086	0.01	6.0	0.986381	0.998439	0.986193	[4469, 14, 0, 500]
30095	0.01	5.0	0.965549	0.995873	0.964320	[4446, 37, 0, 500]
30110	0.01	8.0	0.936347	0.989304	0.932458	[4414, 69, 3, 497]
30118	0.01	7.0	0.817662	0.967990	0.777001	[4196, 287, 0, 500]

Cuadro 4.5: Mejores Configuraciones por Transformador

Transformador	Nu	Noise Factor	PR-AUC	ROC-AUC	F1-Score	CM
10023	0.01	8.0	0.828084	0.971503	0.792393	[4335, 262, 0, 500]
10193	0.01	7.0	0.897660	0.985187	0.886323	[4470, 127, 1, 499]
10292	0.01	8.0	0.908497	0.987818	0.899281	[4485, 112, 0, 500]
10370	0.01	8.0	0.948833	0.993800	0.946074	[4540, 57, 0, 500]
10729	0.01	8.0	0.840229	0.972984	0.811075	[4367, 230, 2, 498]
11031	0.01	7.0	0.821867	0.969198	0.783975	[4323, 274, 1, 499]
11081	0.01	8.0	0.911132	0.986580	0.902992	[4492, 105, 2, 498]
11132	0.01	7.0	0.771739	0.954318	0.704225	[4177, 420, 0, 500]
11139	0.01	8.0	0.906504	0.987492	0.896861	[4482, 115, 0, 500]
11194	0.01	7.0	0.861795	0.979226	0.839631	[4406, 191, 0, 500]
11607	0.01	8.0	0.942478	0.992930	0.938967	[4532, 65, 0, 500]
30051	0.01	7.0	0.832889	0.972700	0.799361	[4346, 251, 0, 500]
30079	0.01	8.0	0.933276	0.991625	0.928505	[4520, 77, 0, 500]
30086	0.01	8.0	0.962107	0.995541	0.960615	[4556, 41, 0, 500]
30095	0.01	7.0	0.813283	0.967588	0.770416	[4299, 298, 0, 500]
30110	0.01	8.0	0.877074	0.982271	0.859845	[4434, 163, 0, 500]
30118	0.01	4.0	0.805998	0.965521	0.759301	[4280, 317, 0, 500]

Cuadro 4.6: Mejores Configuraciones por Transformador (Dataset + 500 datos predichos)

El método **LOF** se evaluó en términos de las mismas métricas. Las Tablas 4.7 y 4.8 presentan los resultados:

Transformador	Contamination	Noise Factor	PR-AUC	ROC-AUC	F1-Score	CM
10023	0.01	7.0	0.946429	0.993308	0.943396	[4423, 60, 0, 500]
10193	0.01	8.0	0.681241	0.763077	0.638498	[4403, 80, 228, 272]
10292	0.01	8.0	0.780728	0.842081	0.760043	[4412, 71, 150, 350]
10370	0.01	5.0	0.957875	0.994870	0.956023	[4437, 46, 0, 500]
10729	0.01	7.0	0.919520	0.987627	0.912924	[4390, 93, 2, 498]
11031	0.01	8.0	0.938909	0.990527	0.935211	[4416, 67, 2, 498]
11081	0.10	8.0	0.822287	0.968552	0.784591	[4210, 273, 1, 499]
11132	0.01	7.0	0.905452	0.986174	0.895871	[4368, 115, 1, 499]
11139	0.01	8.0	0.946429	0.993308	0.943396	[4423, 60, 0, 500]
11194	0.01	8.0	0.940141	0.992416	0.936330	[4415, 68, 0, 500]
11607	0.01	8.0	0.914933	0.973412	0.910293	[4406, 77, 18, 482]
30051	0.01	6.0	0.943262	0.992862	0.939850	[4419, 64, 0, 500]
30079	0.05	7.0	0.863901	0.979143	0.842460	[4296, 187, 0, 500]
30086	0.10	8.0	0.755885	0.946799	0.677048	[4006, 477, 0, 500]
30095	0.05	7.0	0.846260	0.975240	0.818331	[4261, 222, 0, 500]
30110	0.01	8.0	0.863034	0.925966	0.856299	[4402, 81, 65, 435]
30118	0.10	6.0	0.722816	0.930627	0.616523	[3861, 622, 0, 500]

Cuadro 4.7: Mejores Configuraciones por Transformador

Transformador	Contamination	Noise Factor	PR-AUC	ROC-AUC	F1-Score	CM
10023	0.01	7.0	0.889408	0.984555	0.875657	[4455, 142, 0, 500]
10193	0.01	8.0	0.927350	0.990755	0.921659	[4512, 85, 0, 500]
10292	0.01	8.0	0.960405	0.995323	0.958773	[4554, 43, 0, 500]
10370	0.01	5.0	0.964684	0.995867	0.963391	[4559, 38, 0, 500]
10729	0.01	7.0	0.914241	0.986232	0.906934	[4498, 99, 3, 497]
11031	0.01	8.0	0.869826	0.980075	0.850810	[4423, 174, 1, 499]
11081	0.10	8.0	0.729052	0.935263	0.629653	[4011, 586, 1, 499]
11132	0.01	7.0	0.863901	0.979661	0.842460	[4410, 187, 0, 500]
11139	0.01	8.0	0.932526	0.991516	0.927644	[4519, 78, 0, 500]
11194	0.01	7.0	0.891561	0.983317	0.879082	[4462, 135, 2, 498]
11607	0.01	8.0	0.938596	0.992386	0.934579	[4527, 70, 0, 500]
30051	0.01	6.0	0.846260	0.975854	0.818331	[4375, 222, 0, 500]
30079	0.05	7.0	0.849162	0.976506	0.822368	[4381, 216, 0, 500]
30086	0.10	7.0	0.742718	0.942354	0.653595	[4067, 530, 0, 500]
30095	0.05	7.0	0.759605	0.949641	0.683527	[4134, 463, 0, 500]
30110	0.01	8.0	0.931034	0.991299	0.925926	[4517, 80, 0, 500]
30118	0.10	8.0	0.715146	0.927997	0.601685	[3935, 662, 0, 500]

Cuadro 4.8: Mejores Configuraciones por Transformador (Dataset + 500 datos predichos)

El método **Elliptic Envelope** mostró el mejor rendimiento global. Las Tablas 4.9 y 4.10 detallan los resultados obtenidos:

Transformador	Contamination	Noise Factor	PR-AUC	ROC-AUC	F1-Score	CM
10023	0.01	6.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]
10193	0.01	8.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]
10292	0.01	8.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]
10370	0.01	8.0	0.979846	0.997658	0.979432	[4462, 21, 0, 500]
10729	0.01	5.0	0.947036	0.992531	0.944182	[4425, 58, 1, 499]
11031	0.01	8.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]
11081	0.01	7.0	0.956899	0.993870	0.955024	[4437, 46, 1, 499]
11132	0.01	7.0	0.994071	0.999331	0.994036	[4477, 6, 0, 500]
11139	0.01	7.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]
11194	0.01	7.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]
11607	0.01	8.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]
30051	0.01	6.0	0.948833	0.993643	0.946074	[4426, 57, 0, 500]
30079	0.01	6.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]
30086	0.01	5.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]
30095	0.01	4.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]
30110	0.01	8.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]
30118	0.01	7.0	0.843879	0.974682	0.814996	[4256, 227, 0, 500]

Cuadro 4.9: Mejores Configuraciones por Transformador

Transformador	Contamination	Noise Factor	PR-AUC	ROC-AUC	F1-Score	CM
10023	0.01	7.0	0.991159	0.999021	0.991080	[4588, 9, 0, 500]
10193	0.01	8.0	0.978927	0.997607	0.978474	[4575, 22, 0, 500]
10292	0.01	6.0	0.980758	0.996933	0.980354	[4578, 19, 1, 499]
10370	0.01	5.0	0.978011	0.997498	0.977517	[4574, 23, 0, 500]
10729	0.01	5.0	0.946231	0.992583	0.943289	[4538, 59, 1, 499]
11031	0.01	8.0	0.993097	0.999239	0.993049	[4590, 7, 0, 500]
11081	0.01	8.0	0.965453	0.995084	0.964251	[4561, 36, 1, 499]
11132	0.01	6.0	0.985437	0.998369	0.985222	[4582, 15, 0, 500]
11139	0.01	7.0	0.973485	0.996955	0.972763	[4569, 28, 0, 500]
11194	0.01	8.0	0.980769	0.997825	0.980392	[4577, 20, 0, 500]
11607	0.01	7.0	0.976190	0.997281	0.975610	[4572, 25, 0, 500]
30051	0.01	4.0	0.965549	0.995976	0.964320	[4560, 37, 0, 500]
30079	0.01	6.0	1.000000	1.000000	1.000000	[4597, 0, 0, 500]
30086	0.01	5.0	1.000000	1.000000	1.000000	[4597, 0, 0, 500]
30095	0.01	5.0	1.000000	1.000000	1.000000	[4597, 0, 0, 500]
30110	0.01	7.0	0.977099	0.997390	0.976562	[4573, 24, 0, 500]
30118	0.01	7.0	0.798686	0.963346	0.747943	[4260, 337, 0, 500]

Cuadro 4.10: Mejores Configuraciones por Transformador (Dataset + 500 datos predichos)

4.5.1. Comparación de Algoritmos

Finalmente, las Tablas 4.11 y 4.12 se presenta un resumen del promedio de métricas obtenidas por algoritmo, destacando el rendimiento de **Elliptic Envelope** sobre los demás métodos.

Algoritmo	F1-Score	PR-AUC	ROC-AUC
Elliptic Envelope	0.978455	0.980621	0.997160
Isolation Forest	0.956624	0.961001	0.994260
LOF	0.845123	0.867594	0.955058
One-Class SVM	0.892202	0.905383	0.983667

Cuadro 4.11: Promedio de métricas: mejores configuraciones - dataset original.

Algoritmo	F1-Score	PR-AUC	ROC-AUC
Elliptic Envelope	0.966519	0.970050	0.995594
Isolation Forest	0.920749	0.930148	0.989818
LOF	0.835063	0.866205	0.975212
One-Class SVM	0.851755	0.874320	0.979781

Cuadro 4.12: Promedio de métricas: mejores configuraciones - datos aumentados.

Finalmente, enfocado por transformador, se obtuvo el mejor algoritmo en cada caso, para el dataset original y para el dataset aumentado:

Transformador	Contamination	Noise Factor	PR-AUC	ROC-AUC	F1-Score	CM	Algoritmo
10023	0.01	6.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]	Elliptic Envelope
10193	0.01	8.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]	Elliptic Envelope
10292	0.01	8.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]	Elliptic Envelope
10370	0.01	6.0	0.983559	0.998104	0.983284	[4466, 17, 0, 500]	Isolation Forest
10729	0.01	5.0	0.947036	0.992531	0.944182	[4425, 58, 1, 499]	Elliptic Envelope
11031	0.01	8.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]	Elliptic Envelope
11081	0.01	7.0	0.962107	0.995427	0.960615	[4442, 41, 0, 500]	Isolation Forest
11132	0.01	7.0	0.994071	0.999331	0.994036	[4477, 6, 0, 500]	Elliptic Envelope
11139	0.01	8.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]	Isolation Forest
11194	0.01	7.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]	Elliptic Envelope
11607	0.01	8.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]	Isolation Forest
30051	0.01	6.0	0.948833	0.993643	0.946074	[4426, 57, 0, 500]	Elliptic Envelope
30079	0.01	6.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]	Elliptic Envelope
30086	0.01	5.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]	Isolation Forest
30095	0.01	4.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]	Isolation Forest
30110	0.01	8.0	1.000000	1.000000	1.000000	[4483, 0, 0, 500]	Elliptic Envelope
30118	0.01	7.0	0.843879	0.974682	0.814996	[4256, 227, 0, 500]	Elliptic Envelope

Transformador	Contamination	Noise Factor	PR-AUC	ROC-AUC	F1-Score	CM	Algoritmo	Nu
10023	0.01	7.0	0.991159	0.999021	0.991080	[4588, 9, 0, 500]	Elliptic Envelope	NaN
10193	0.01	8.0	0.978927	0.997607	0.978474	[4575, 22, 0, 500]	Elliptic Envelope	NaN
10292	0.01	7.0	0.980769	0.997825	0.980392	[4577, 20, 0, 500]	Isolation Forest	NaN
10370	0.01	4.0	0.988314	0.997804	0.988119	[4586, 11, 1, 499]	Isolation Forest	NaN
10729	0.01	5.0	0.946231	0.992583	0.943289	[4538, 59, 1, 499]	Elliptic Envelope	NaN
11031	0.01	8.0	0.993097	0.999239	0.993049	[4590, 7, 0, 500]	Elliptic Envelope	NaN
11081	0.01	7.0	0.965453	0.995084	0.964251	[4561, 36, 1, 499]	Isolation Forest	NaN
11132	0.01	6.0	0.985437	0.998369	0.985222	[4582, 15, 0, 500]	Elliptic Envelope	NaN
11139	0.01	7.0	0.973485	0.996955	0.972763	[4569, 28, 0, 500]	Elliptic Envelope	NaN
11194	0.01	8.0	0.980769	0.997825	0.980392	[4577, 20, 0, 500]	Elliptic Envelope	NaN
11607	0.01	7.0	0.976190	0.997281	0.975610	[4572, 25, 0, 500]	Elliptic Envelope	NaN
30051	0.01	4.0	0.965549	0.995976	0.964320	[4560, 37, 0, 500]	Elliptic Envelope	NaN
30079	0.01	6.0	1.000000	1.000000	1.000000	[4597, 0, 0, 500]	Elliptic Envelope	NaN
30086	0.01	5.0	1.000000	1.000000	1.000000	[4597, 0, 0, 500]	Isolation Forest	NaN
30095	0.01	5.0	1.000000	1.000000	1.000000	[4597, 0, 0, 500]	Isolation Forest	NaN
30110	0.01	7.0	0.977099	0.997390	0.976562	[4573, 24, 0, 500]	Elliptic Envelope	NaN
30118	NaN	4.0	0.805998	0.965521	0.759301	[4280, 317, 0, 500]	One-Class SVM	0.01

Conclusiones

Conclusiones sobre Modelos Predictivos

- **LSTM como el mejor modelo predictivo:** Su capacidad para capturar patrones temporales y cíclicos lo hace ideal para series temporales como la demanda energética en transformadores eléctricos. Su rendimiento consistente lo posiciona como la mejor alternativa frente a modelos más complejos como los híbridos.
- Aunque los modelos híbridos (como la combinación de TCN y LSTM) ofrecen un rendimiento competitivo, los beneficios marginales obtenidos no justifican su mayor complejidad frente a un LSTM optimizado. Por esta razón, se recomienda utilizar exclusivamente LSTM para tareas futuras de predicción, especialmente en configuraciones MIMO.
- **Transformers** y **TCN** mostraron limitaciones frente al LSTM, con métricas de rendimiento significativamente menores, lo que sugiere que no son adecuados para este dominio específico.

Conclusiones sobre la Detección de Anomalías

- **Elliptic Envelope** destacó como el mejor modelo para la detección de anomalías, especialmente en el dataset original. Esto se debe a su capacidad para capturar distribuciones gaussianas en datos multivariantes y su alineación con el comportamiento cíclico regular de los transformadores eléctricos.
- Los datos aumentados mediante predicciones del modelo LSTM mantuvieron estadísticas globales similares a las del dataset original. Esto permitió que

los algoritmos de detección de anomalías, como Isolation Forest y Elliptic Envelope, preservaran su rendimiento.

- Los experimentos con factores de ruido y niveles de contaminación mostraron que el rendimiento de los modelos se degrada de manera proporcional a la magnitud del ruido, pero Elliptic Envelope y Isolation Forest mantuvieron una robustez notable incluso en estas condiciones.
- **LOF y One-Class SVM**, aunque útiles, demostraron un desempeño menos robusto comparado con Elliptic Envelope e Isolation Forest, particularmente con datos aumentados o con mayores niveles de ruido.

Recomendaciones y Trabajos Futuros

- Explorar métodos híbridos que combinen algoritmos supervisados y no supervisados para mejorar la detección de anomalías.
- Analizar cómo diferentes arquitecturas predictivas, como Transformers, afectan la generación de datos sintéticos y su impacto en la detección de anomalías.
- Evaluar el efecto de introducir diferentes tipos de anomalías (como cambios abruptos o anomalías contextualizadas) para simular escenarios más complejos.

Conclusión General

El modelo LSTM se consolida como la herramienta más eficiente y confiable para la predicción de series temporales en transformadores eléctricos, mientras que Elliptic Envelope y Isolation Forest destacan en la detección de anomalías debido a su robustez frente a ruido y datos aumentados. Estos resultados demuestran que la integración de predicción y detección de anomalías es una solución viable y efectiva para el monitoreo y mantenimiento de infraestructuras críticas como los transformadores eléctricos.

Bibliografía

- [1] Ankle, L. L., Hegglund, M. F., & Krange, K. (2020). “Deep Convolutional Neural Networks: A survey of the foundations, selected improvements, and some current applications.” *arXiv preprint arXiv:2011.12960*. Recuperado de <https://arxiv.org/abs/2011.12960>
- [2] Biewald, L. (2020). “Experiment Tracking with Weights and Biases.” Wandb. Recuperado de <https://www.wandb.com/>
- [3] Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). “LOF: Identifying Density-Based Local Outliers.” In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (pp. 93–104). ACM. <https://doi.org/10.1145/342009.335388>
- [4] Cayci, S., & Eryilmaz, A. (2024). “Convergence of Gradient Descent for Recurrent Neural Networks: A Nonasymptotic Analysis.” *arXiv preprint arXiv:2402.12241*. Recuperado de <https://arxiv.org/abs/2402.12241>
- [5] DeepMind. (2022). “WaveNet: A generative model for raw audio.” Recuperado de <https://deepmind.google/discover/blog/wavenet-a-generative-model-for-raw-audio>
- [6] Li, H., Zhang, Z., & Liu, J. (2019). “Detection of False Data Injection Attacks in Smart Grids Using Game Theory.” *International Journal of Electrical Power & Energy Systems*, 109, 575-581.
- [7] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). “Isolation Forest.” In *Proceedings of the 8th IEEE International Conference on Data Mining* (pp. 413–422). IEEE. <https://doi.org/10.1109/ICDM.2008.17>
- [8] O’Shea, K., & Nash, R. (2015). “An Introduction to Convolutional Neural Networks.” *arXiv preprint arXiv:1511.08458*. Recuperado de <https://arxiv.org/abs/1511.08458>

- [9] Rousseeuw, P. J., & Driessen, K. V. (1999). “A Fast Algorithm for the Minimum Covariance Determinant Estimator.” *Technometrics*, 41(3), 212–223. <https://doi.org/10.1080/00401706.1999.10485670>
- [10] Schuster, M., & Paliwal, K. K. (1997). “Bidirectional recurrent neural networks.” *IEEE Transactions on Signal Processing*, 45(11), 2673–2681.
- [11] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). “Estimating the Support of a High-Dimensional Distribution.” *Neural Computation*, 13(7), 1443–1471. <https://doi.org/10.1162/089976601750264965>
- [12] Shervine, A., et al. (2021). “Custom Data Generation for Keras Models.” Recuperado de <https://medium.com/analytics-vidhya/write-your-own-custom-data-generator-for-tensorflow-keras-1252b64e41c3>
- [13] Staudemeyer, R. C., & Morris, E. R. (2019). “Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks.” *arXiv preprint arXiv:1909.09586*. Recuperado de <https://arxiv.org/abs/1909.09586>
- [14] Stanford University. (2021). “Data generation in Keras.” Recuperado de <https://stanford.edu/~shervine/blog/keras-how-to-generate-data-on-the-fly>
- [15] Stanford University. (2024). “Recurrent Neural Networks cheatsheet.” Recuperado de <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>
- [16] Wang, X., et al. (2022). “A Survey on Transformer-Based Machine Learning for Energy Demand Forecasting.” *Energy and AI*, 3, 100056.
- [17] Zhang, Y., Wang, X., & Liu, J. (2020). “Detecting False Data Injection Attacks in Smart Grids Using CNNs and LSTMs.” *IEEE Transactions on Smart Grid*, 11(4), 3043–3051. <https://arxiv.org/pdf/2006.11477>

Anexos

Se aclara que todos los programas mencionados en este proyecto pueden ser encontrados y accedidos libremente a través del siguiente repositorio de Github: <https://github.com/Negatix092/FP.git>

Anexo A: Figuras sobre FFT y PSD para distintos Transformadores

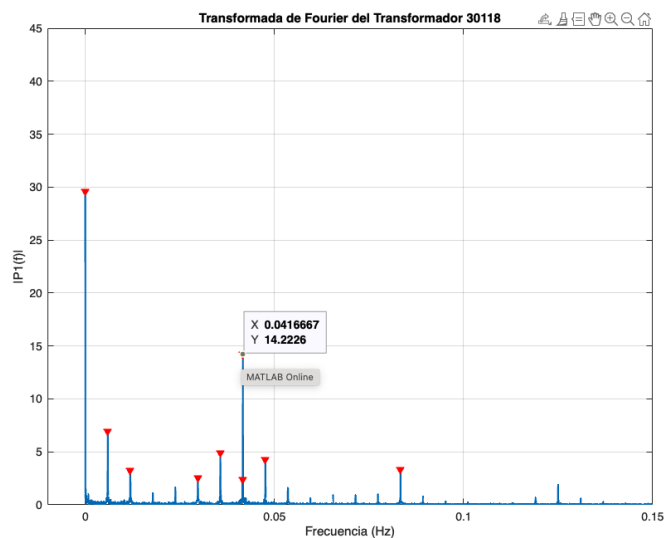


Figura 4.6: Transformada de Fourier (FFT) en el Dominio de Frecuencias para el Transformador 30118

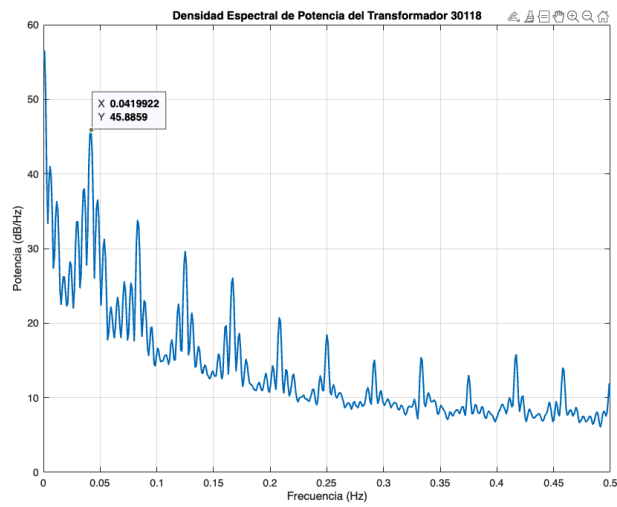


Figura 4.7: Densidad Espectral de Potencia (PSD) del Transformador 30118, donde se destacan las frecuencias dominantes.

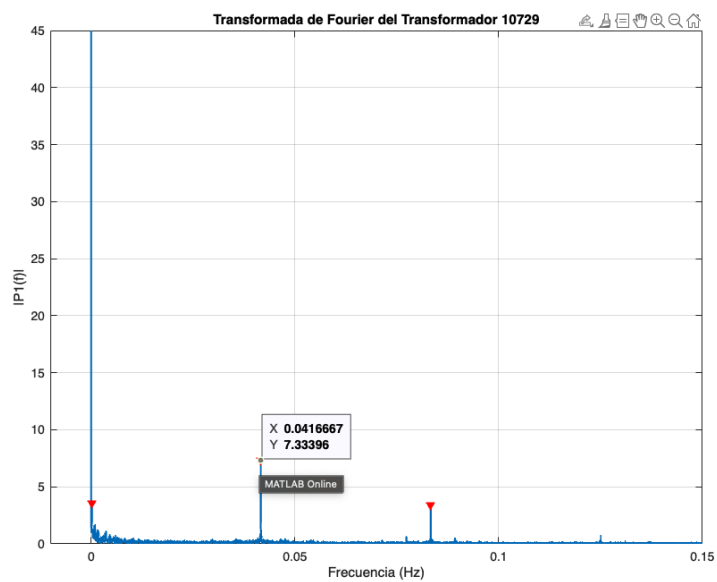


Figura 4.8: Transformada de Fourier (FFT) en el Dominio de Frecuencias para el Transformador 10729

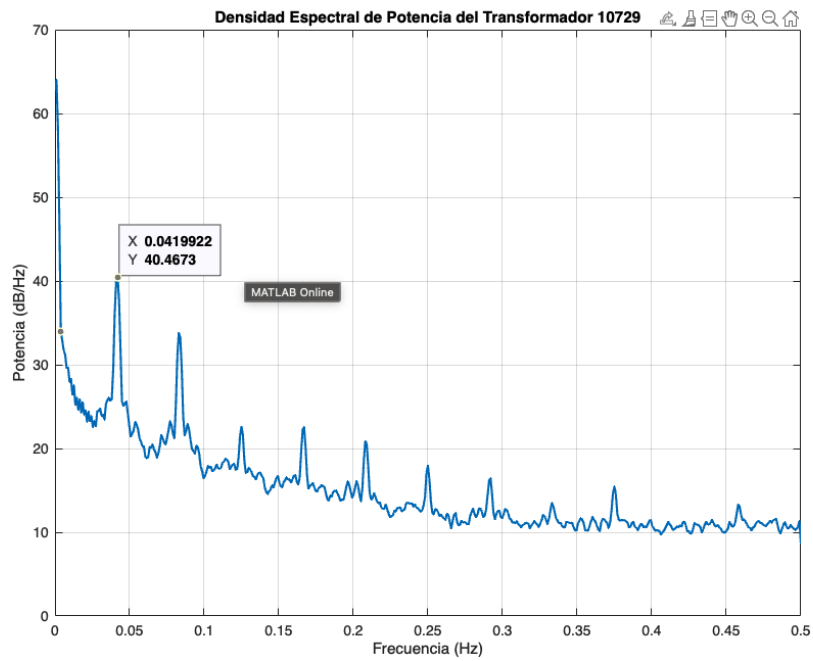


Figura 4.9: Densidad Espectral de Potencia (PSD) del Transformador 10729, donde se destacan las frecuencias dominantes.