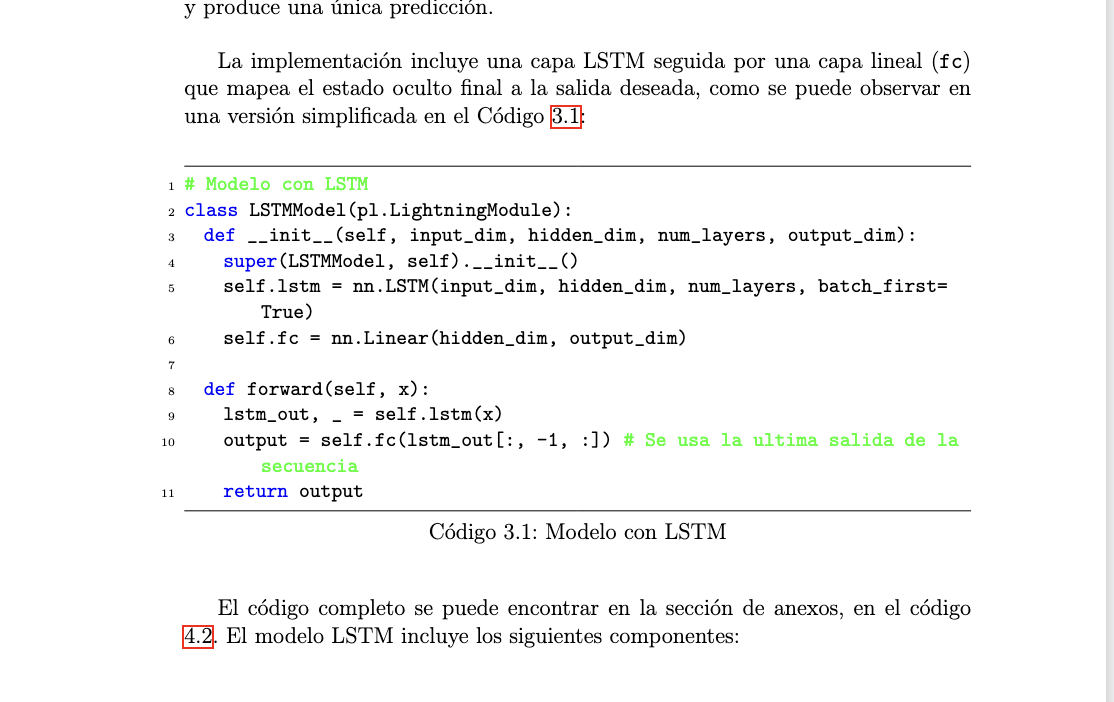
Para construir un **intervalo de confianza** en base a las predicciones de tu modelo y al MSE obtenido, primero es necesario entender cómo se calculan estos intervalos. Un intervalo de confianza para las predicciones refleja el rango en el cual esperamos que caigan los valores reales en función de las predicciones del modelo, dado un nivel de confianza (por ejemplo, 95%).

**Teoría del Intervalo de Confianza:**

* Para una predicción y^iy^​i​, el intervalo de confianza (IC) a un nivel de confianza αα generalmente sigue esta fórmula:IC=y^i±Zα/2⋅MSEnIC=y^​i​±Zα/2​⋅nMSE​​ Donde:
  + y^iy^​i​ es el valor predicho.
  + Zα/2Zα/2​ es el valor crítico de una distribución normal estándar (por ejemplo, 1.96 para un intervalo del 95%).
  + MSEMSE es el error cuadrático medio obtenido en el conjunto de validación.
  + nn es el número de observaciones o la ventana sobre la cual calculas los intervalos.

Luego de imagen 3.1



El modelo LSTM incluye los siguientes componentes:

\begin{itemize}

\item \textbf{Capa LSTM:} Definida como \texttt{self.lstm}, donde \texttt{input\\_dim} corresponde al número de características en cada paso de tiempo (en este caso, es 1), \texttt{hidden\\_dim} define el tamaño de las celdas de memoria ocultas, y \texttt{num\\_layers} indica la cantidad de capas LSTM utilizadas. Esta capa permite que la secuencia de datos fluya a través del tiempo y pueda aprender dependencias temporales a largo plazo.

\item \textbf{Capa Lineal:} \texttt{self.fc} toma como entrada el estado oculto final de la secuencia (última salida) y lo transforma en la salida deseada. En este caso, \texttt{output\\_dim} corresponde a una única predicción.

\item \textbf{Método \texttt{forward()}:} Define cómo el modelo procesa los datos de entrada. Primero, se aplica la capa LSTM sobre la secuencia de entrada y, luego, la salida oculta final (\texttt{lstm\\_out[:, -1, :]}) se pasa a la capa lineal (\texttt{fc}) para generar la predicción. Esto permite que el modelo tome en cuenta toda la información acumulada durante la secuencia para realizar una predicción precisa.

\end{itemize}

Para encontrar el tamaño de ventana adecuado que permita capturar la periodicidad de las señales de demanda energética de manera experimental, se evaluaron diferentes tamaños de ventana en un rango de 1 a 48, que corresponden a horas. Este proceso se realizó mediante una función (\texttt{evaluate\\_window\\_size}) que crea un \texttt{DataModule} con los datos de entrenamiento, validación y prueba. La implementación de esta función se puede ver en el Código \ref{lst:evaluate\_window}:

\vspace{0.5 cm}

\begin{lstlisting}[language=Python, caption=Evaluar el tamaño de la ventana, label={lst:evaluate\_window}]

# Evaluar el tamaño de la ventana

def evaluate\_window\_size(train\_data, val\_data, test\_data, window\_size):

data\_module = DemandDataModule(train\_data, val\_data, test\_data, window\_size)

input\_dim = 1

hidden\_dim = 64

num\_layers = 2

output\_dim = 1

model = LSTMModel(input\_dim=input\_dim, hidden\_dim=hidden\_dim, num\_layers=num\_layers, output\_dim=output\_dim)

# Definir el entrenador

trainer = pl.Trainer(max\_epochs=10, logger=False, enable\_checkpointing=False)

# Entrenar el modelo

trainer.fit(model, data\_module)

# Acceder a las métricas después del entrenamiento

train\_loss = trainer.callback\_metrics.get('train\_loss')

val\_result = trainer.validate(model, data\_module)

return train\_loss.item() if train\_loss else None, val\_result[0]['val\_loss'], model, data\_module

\end{lstlisting}

El usar capas recurrentes para procesar la secuencia de datos significa procesar información de manera secuencial para retener dependencias a largo plazo. Este proceso puede limitar la cantidad de capas debido a problemas de "desvanecimiento del gradiente", pues, a pesar de que al aumentar el número de capas se tiene una mayor capacidad para aprender dependencias temporales complejas \cite{staudemeyer2019}, agregar muchas puede traducirse en problemas de overfitting (sobreajuste) y altos costos computacionales. Es por esto que, para mantener un equilibrio entre capacidad de modelado y eficiencia, se ha optado por 2 capas.

El parámetro hidden$\\_$dim = 64 hace referencia al tamaño de las celdas de memoria internas que permiten aprender las dependencias temporales. Este valor fue considerado como un tamaño suficientemente representativo para la complejidad del problema.

A screenshot of a computer program

Description automatically generated

El modelo TCN incluye los siguientes componentes:

\begin{itemize}

\item \textbf{Capas Convolucionales con Dilatación:} Definidas en un bucle que crea tantas capas como \texttt{num\\_layers} especificados. Cada capa aplica convoluciones 1D con un \texttt{kernel\\_size} determinado y un \texttt{dilation\\_size} que se incrementa exponencialmente (\texttt{2 \*\* i}). Esto permite a TCN capturar dependencias a largo plazo sin requerir un procesamiento secuencial.

\item \textbf{Activación y Dropout:} Después de cada convolución se aplica una función de activación ReLU seguida de una capa de Dropout para evitar el sobreajuste. Esto se realiza mediante \texttt{nn.ReLU()} y \texttt{nn.Dropout(dropout)}.

\item \textbf{Capa Lineal:} Similar al modelo LSTM, la capa lineal \texttt{self.fc} toma la salida del último paso de la secuencia para generar la predicción.

\item \textbf{Método \texttt{forward()}:} Dado que la convolución 1D en PyTorch espera entradas con la forma (batch\\_size, channels, sequence\\_length), es necesario transponer la entrada antes de aplicar la TCN y después del procesamiento convolucional (\texttt{x = x.transpose(1, 2)} y \texttt{tcn\\_out = tcn\\_out.transpose(1, 2)}). La salida luego pasa por la capa lineal para obtener la predicción final.

\end{itemize}

Similar a la implementación del modelo LSTM, se procedió a evaluar diferentes tamaños de ventana con el modelo TCN, como se puede observar en el Código \ref{lst:evaluate\_tcn\_window}.

\vspace{0.5cm}

\begin{lstlisting}[language=Python, caption=Evaluar el tamaño de la ventana, label={lst:evaluate\_tcn\_window}]

# Evaluar el tamano de la ventana

def evaluate\_window\_size(train\_data, val\_data, test\_data, window\_size):

data\_module = DemandDataModule(train\_data, val\_data, test\_data, window\_size)

input\_dim = 1 # Cada valor de entrada es un valor por paso de tiempo

hidden\_dim = 64 # Numero de canales ocultos de la TCN

num\_layers = 3 # Numero de capas convolucionales

output\_dim = 1 # Queremos predecir un solo valor (proxima demanda)

# Instancia el modelo TCN

model = TCNModel(input\_dim=input\_dim, hidden\_dim=hidden\_dim, output\_dim=output\_dim, num\_layers=num\_layers)

\end{lstlisting}

En TCN, el aumento de capas ayuda a capturar dependencias de largo alcance y, gracias a su capacidad para realizar operaciones en paralelo, agregar más capas resulta más eficiente que en el caso de LSTM \cite{ankle2020, oshea2015}. Por eso se usa un num$\\_$layers = 3, aumentando su capacidad de representar características espaciales y temporales sin los problemas comunes de las RNN como el desvanecimiento del gradiente.

El modelo TCN fue diseñado para capturar dependencias temporales, tanto a corto como a largo plazo, en una secuencia de datos. Una de las principales características del modelo TCN es el uso de \textbf{convoluciones dilatadas}, que permiten que la red pueda tener una ventana de percepción más amplia con el aumento de capas, sin necesidad de aumentar el número de parámetros excesivamente \cite{oshea2015}.

\begin{itemize}

\item \textbf{Capas Convolucionales con Dilatación:}

Las capas convolucionales de las TCN están diseñadas para aplicar convoluciones sobre los datos, lo que implica realizar operaciones que permiten extraer características específicas de la secuencia \cite{ankle2020}. En el modelo TCN, se utilizan convoluciones \textbf{1D}, ya que se trabaja con datos unidimensionales (como las series temporales).

En el Código \ref{lst:tcn\_model}, se puede observar que las capas convolucionales se definen dentro de un bucle. Este bucle se ejecuta \texttt{num\\_layers} veces, con el objetivo de construir múltiples capas para capturar características más complejas de la secuencia. Aquí es importante entender la noción de \textbf{dilatación}:

\begin{itemize}

\item La \textbf{dilatación} se refiere a la separación entre los valores sobre los que se aplica la convolución. Por ejemplo, una dilatación de 1 significa que cada valor es adyacente al siguiente, mientras que una dilatación de 2 significa que se omite un valor entre cada aplicación. En el código, el valor de dilatación se establece como \texttt{2 \*\* i}, donde \texttt{i} es el número de la capa, lo cual hace que la dilatación aumente exponencialmente (1, 2, 4, etc.). Esto permite que la red capture relaciones a largo plazo de manera más eficiente.

\item La \textbf{convolución dilatada} permite que la TCN tenga una ventana de percepción mayor sin necesidad de aumentar mucho la cantidad de parámetros o el tamaño del kernel. Es decir, el modelo puede ver una mayor porción de la secuencia a medida que aumenta el número de capas y la dilatación, lo cual resulta fundamental para capturar patrones de larga duración en las series temporales \cite{ankle2020}.

\end{itemize}

En cada capa, se especifica un \texttt{padding\\_size} que garantiza que la salida tenga el mismo tamaño que la entrada, lo cual es útil para que las diferentes capas convolucionales se puedan encadenar sin problemas.

\item \textbf{Activación y Dropout:}

Después de cada operación de convolución, se aplica una \textbf{función de activación ReLU} (\texttt{nn.ReLU()}) para introducir no linealidades en el modelo. Esto es importante para que el modelo pueda aprender relaciones complejas y no lineales en los datos.

Luego, se aplica una \textbf{capa de Dropout} (\texttt{nn.Dropout(dropout)}) con el fin de evitar el sobreajuste del modelo. El Dropout desactiva aleatoriamente algunas de las unidades durante el entrenamiento, lo cual obliga al modelo a ser más robusto y a no depender demasiado de características específicas. El valor de \texttt{dropout} se suele establecer en un valor bajo (en este caso, 0.2) para evitar la desactivación excesiva de unidades.

\item \textbf{Capa Lineal:}

Similar al modelo LSTM, la \textbf{capa lineal} (\texttt{self.fc}) toma la salida del último paso de la secuencia para generar la predicción. Esta capa es la responsable de convertir las características extraídas por las capas convolucionales en un valor de salida que corresponda a la predicción deseada, como la demanda energética para el próximo período.

\item \textbf{Método \texttt{forward()}:}

La función \texttt{forward()} define cómo se procesan los datos en el modelo. Es importante notar que, dado que la convolución 1D en PyTorch espera que los datos de entrada tengan la forma \texttt{(batch\\_size, channels, sequence\\_length)}, es necesario transponer la entrada (\texttt{x = x.transpose(1, 2)}) antes de aplicar la TCN y, luego de aplicar las capas convolucionales, se vuelve a transponer la salida (\texttt{tcn\\_out = tcn\\_out.transpose(1, 2)}).

Finalmente, la salida se pasa por la capa lineal para obtener la predicción final de la serie temporal.

\end{itemize}

A screenshot of a computer program

Description automatically generated

El modelo híbrido incluye los siguientes componentes:

\begin{itemize}

\item \textbf{Capas Convolucionales con Dilatación (TCN):} Definidas mediante \texttt{nn.Conv1d}, estas capas se encargan de capturar patrones temporales mediante el uso de convoluciones con dilatación exponencial (\texttt{2 \*\* i}). Esta dilatación permite que el modelo observe relaciones a mayor distancia temporal sin necesidad de aumentar el número de parámetros. Las capas TCN están dispuestas de manera secuencial (\texttt{nn.Sequential}) y cada capa está seguida por una función de activación ReLU y un Dropout para evitar el sobreajuste.

\item \textbf{Capa LSTM:} La salida de la TCN se alimenta a una capa LSTM (\texttt{self.lstm}) que se encarga de aprender dependencias secuenciales en las características extraídas por la TCN. La LSTM se configura con un \texttt{hidden\\_size} de 64, que permite manejar información a nivel secuencial.

\item \textbf{Capa Lineal (Fully Connected):} Finalmente, la capa de salida (\texttt{self.fc}) convierte la última salida de la LSTM en la predicción requerida. En este caso, se predice un solo valor que corresponde a la demanda de energía futura.

\item \textbf{Método \texttt{forward()}:} El método \texttt{forward()} define cómo los datos fluyen a través del modelo. Primero, los datos pasan por la TCN, luego la salida se transfiere a la LSTM, y finalmente la última salida de la secuencia de la LSTM se convierte en la predicción mediante la capa lineal.

\end{itemize}

El modelo híbrido se diseñó con la intención de combinar la capacidad de la TCN de capturar patrones a largo plazo mediante convoluciones con la capacidad de la LSTM de manejar dependencias secuenciales complejas. Esto permite al modelo captar de manera más robusta las variaciones en la demanda de energía.

La función \texttt{evaluate\\_window\\_size} se utiliza para probar diferentes tamaños de ventana con el objetivo de identificar el que mejor captura la periodicidad de la demanda de energía. Esta función crea un módulo de datos (\texttt{DemandDataModule}) con los conjuntos de datos de entrenamiento, validación y prueba. La implementación de esta función se puede observar en el Código \ref{lst:evaluate\_hybrid\_window}:

\vspace{0.5cm}

\begin{lstlisting}[language=Python, caption=Evaluar el tamaño de la ventana en el modelo híbrido, label={lst:evaluate\_hybrid\_window}]

# Evaluar el tamaño de la ventana

def evaluate\_window\_size(train\_data, val\_data, test\_data, window\_size):

data\_module = DemandDataModule(train\_data, val\_data, test\_data, window\_size)

input\_dim = 1 # Cada valor de entrada es un valor por paso de tiempo

tcn\_hidden\_dim = 64 # Número de canales ocultos de la TCN

lstm\_hidden\_dim = 64 # Dimensión oculta de la LSTM

num\_tcn\_layers = 3 # Número de capas convolucionales

num\_lstm\_layers = 1 # Número de capas de la LSTM

output\_dim = 1 # Queremos predecir un solo valor (próxima demanda)

# Instancia el modelo híbrido

model = HybridModel(input\_dim=input\_dim, tcn\_hidden\_dim=tcn\_hidden\_dim, lstm\_hidden\_dim=lstm\_hidden\_dim, output\_dim=output\_dim, num\_tcn\_layers=num\_tcn\_layers)

# Definir el entrenador

trainer = pl.Trainer(max\_epochs=10, logger=False, enable\_checkpointing=False)

# Entrenar el modelo

trainer.fit(model, data\_module)

# Acceder a las métricas después del entrenamiento

train\_loss = trainer.callback\_metrics.get('train\_loss')

val\_result = trainer.validate(model, data\_module)

return train\_loss.item() if train\_loss else None, val\_result[0]['val\_loss'], model, data\_module

\end{lstlisting}

En la función anterior se establecen ciertos valores específicos para la configuración del modelo:

\begin{itemize}

\item \textbf{\texttt{input\\_dim} = 1:} Cada valor de entrada es un único valor de la serie temporal (es decir, la demanda energética en un instante de tiempo).

\item \textbf{\texttt{tcn\\_hidden\\_dim} = 64:} Esta es la dimensión oculta de las capas convolucionales (TCN). Se seleccionó un valor de 64 como un compromiso entre capacidad de representación y eficiencia computacional.

\item \textbf{\texttt{lstm\\_hidden\\_dim} = 64:} Similar al \texttt{tcn\\_hidden\\_dim}, se seleccionó 64 como tamaño del vector de estado oculto de la LSTM para garantizar la consistencia en las capacidades de aprendizaje.

\item \textbf{\texttt{num\\_tcn\\_layers} = 3:} Este valor indica el número de capas convolucionales que se aplican secuencialmente. Se eligieron 3 capas para permitir una mayor profundidad y capacidad de aprendizaje.

\item \textbf{\texttt{num\\_lstm\\_layers} = 1:} Se utilizó una única capa LSTM debido a la naturaleza secuencial de los datos. Esto es suficiente para capturar la relación temporal restante después del procesamiento inicial por las capas convolucionales.

\item \textbf{\texttt{output\\_dim} = 1:} El modelo predice un único valor que representa la demanda energética en el siguiente instante de tiempo.

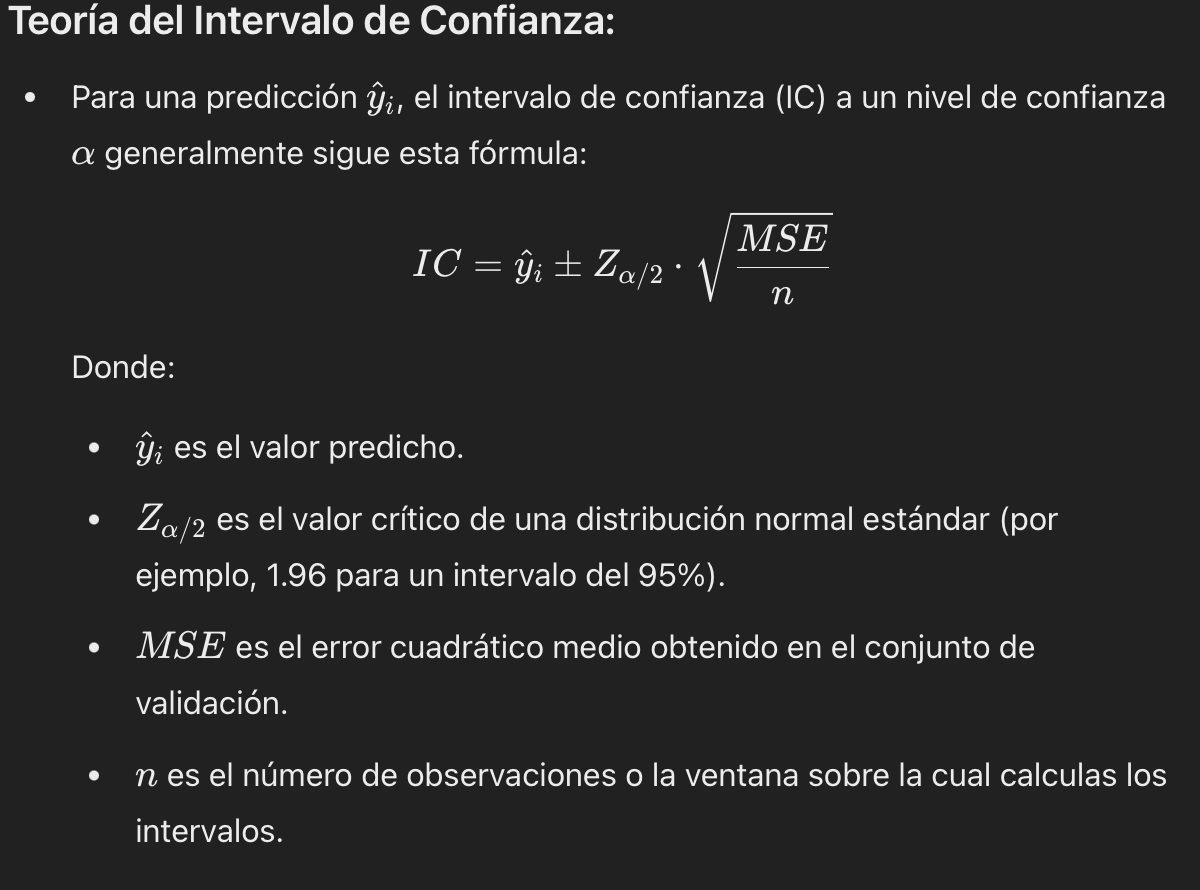
\end{itemize}

\noindent

La función \texttt{evaluate\\_window\\_size} permite entrenar el modelo con diferentes tamaños de ventana y registrar las métricas de entrenamiento y validación para cada uno, lo cual es crucial para determinar el tamaño de ventana más efectivo para capturar patrones cíclicos en los datos de demanda energética.

Los transformers permiten enfocar el análisis en aquellas secciones de la secuencia que tienen mayor influencia sobre el comportamiento futuro, lo cual es particularmente útil para identificar anomalías.

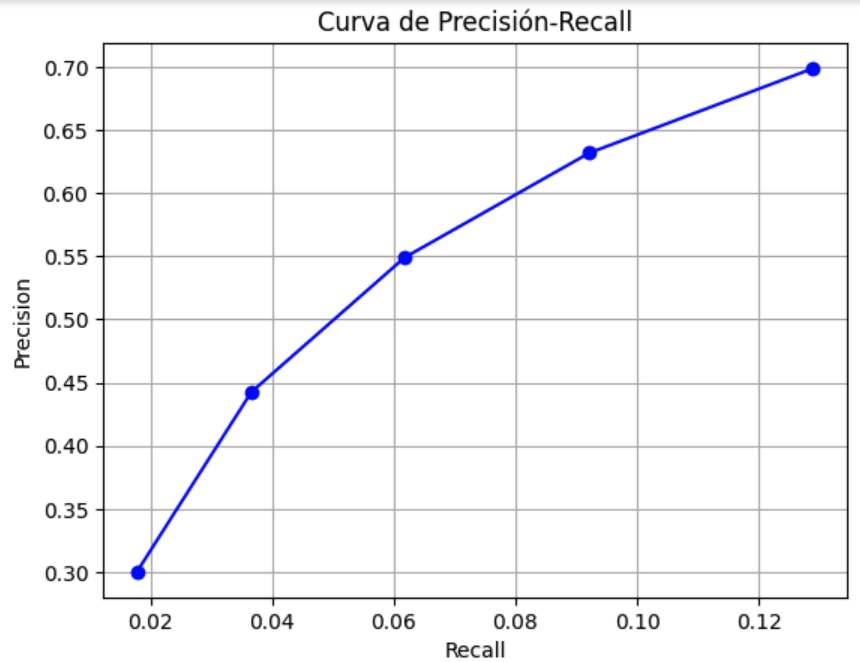
Un intervalo de confianza para las predicciones refleja el rango en el cual esperamos que caigan los valores reales en función de las predicciones del modelo, dado un nivel de confianza (por ejemplo, 95%).

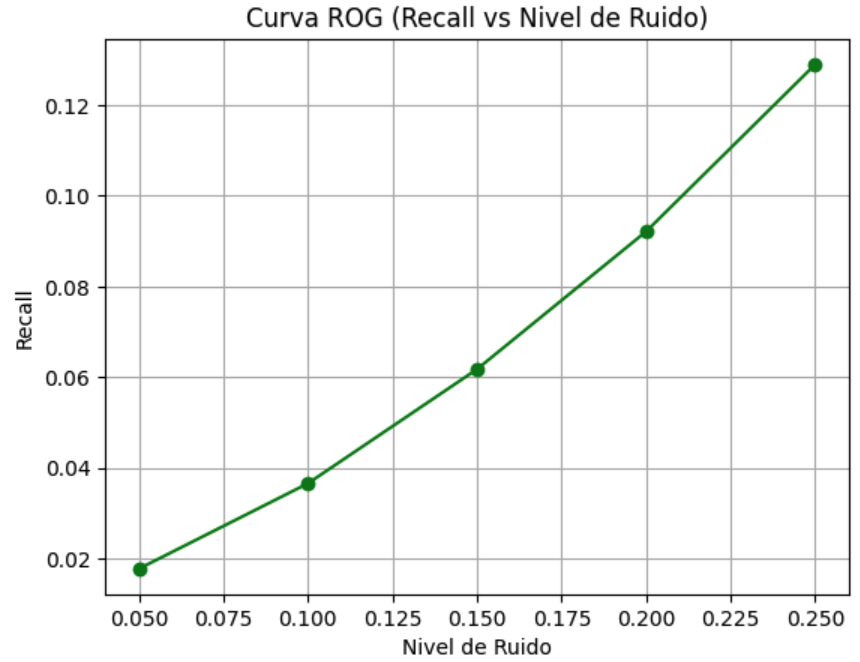


7 de noviembre

RESULTADOS

=== Resultados usando factor de ruido: 0.05 === Transformador 1 Matriz de Confusión: [[ 6 494] [ 18 1958]] Precisión: 0.2500 Recall: 0.0120 Porcentaje de anomalías detectadas: 1.20% Transformador 2 Matriz de Confusión: [[ 5 495] [ 22 1954]] Precisión: 0.1852 Recall: 0.0100 Porcentaje de anomalías detectadas: 1.00% Transformador 3 Matriz de Confusión: [[ 4 496] [ 22 1954]] Precisión: 0.1538 Recall: 0.0080 Porcentaje de anomalías detectadas: 0.80% Transformador 4 Matriz de Confusión: [[ 11 489] [ 21 1955]] Precisión: 0.3438 Recall: 0.0220 Porcentaje de anomalías detectadas: 2.20% Transformador 5 Matriz de Confusión: [[ 15 485] [ 22 1954]] Precisión: 0.4054 Recall: 0.0300 Porcentaje de anomalías detectadas: 3.00% Transformador 6 Matriz de Confusión: [[ 6 494] [ 19 1957]] Precisión: 0.2400 Recall: 0.0120 Porcentaje de anomalías detectadas: 1.20% Transformador 7 Matriz de Confusión: [[ 14 486] [ 16 1960]] Precisión: 0.4667 Recall: 0.0280 Porcentaje de anomalías detectadas: 2.80% Transformador 8 Matriz de Confusión: [[ 8 492] [ 17 1959]] Precisión: 0.3200 Recall: 0.0160 Porcentaje de anomalías detectadas: 1.60% Transformador 9 Matriz de Confusión: [[ 7 493] [ 23 1953]] Precisión: 0.2333 Recall: 0.0140 Porcentaje de anomalías detectadas: 1.40% Transformador 10 Matriz de Confusión: [[ 5 495] [ 23 1953]] Precisión: 0.1786 Recall: 0.0100 Porcentaje de anomalías detectadas: 1.00% Transformador 11 Matriz de Confusión: [[ 5 495] [ 23 1953]] Precisión: 0.1786 Recall: 0.0100 Porcentaje de anomalías detectadas: 1.00% Transformador 12 Matriz de Confusión: [[ 14 486] [ 13 1963]] Precisión: 0.5185 Recall: 0.0280 Porcentaje de anomalías detectadas: 2.80% Transformador 13 Matriz de Confusión: [[ 14 486] [ 18 1958]] Precisión: 0.4375 Recall: 0.0280 Porcentaje de anomalías detectadas: 2.80% Transformador 14 Matriz de Confusión: [[ 11 489] [ 19 1957]] Precisión: 0.3667 Recall: 0.0220 Porcentaje de anomalías detectadas: 2.20% Transformador 15 Matriz de Confusión: [[ 15 485] [ 17 1959]] Precisión: 0.4688 Recall: 0.0300 Porcentaje de anomalías detectadas: 3.00% Transformador 16 Matriz de Confusión: [[ 7 493] [ 19 1957]] Precisión: 0.2692 Recall: 0.0140 Porcentaje de anomalías detectadas: 1.40% Transformador 17 Matriz de Confusión: [[ 20 480] [ 21 1955]] Precisión: 0.4878 Recall: 0.0400 Porcentaje de anomalías detectadas: 4.00% === Resultados usando factor de ruido: 0.1 === Transformador 1 Matriz de Confusión: [[ 17 483] [ 18 1958]] Precisión: 0.4857 Recall: 0.0340 Porcentaje de anomalías detectadas: 3.40% Transformador 2 Matriz de Confusión: [[ 9 491] [ 22 1954]] Precisión: 0.2903 Recall: 0.0180 Porcentaje de anomalías detectadas: 1.80% Transformador 3 Matriz de Confusión: [[ 7 493] [ 22 1954]] Precisión: 0.2414 Recall: 0.0140 Porcentaje de anomalías detectadas: 1.40% Transformador 4 Matriz de Confusión: [[ 15 485] [ 21 1955]] Precisión: 0.4167 Recall: 0.0300 Porcentaje de anomalías detectadas: 3.00% Transformador 5 Matriz de Confusión: [[ 43 457] [ 22 1954]] Precisión: 0.6615 Recall: 0.0860 Porcentaje de anomalías detectadas: 8.60% Transformador 6 Matriz de Confusión: [[ 11 489] [ 19 1957]] Precisión: 0.3667 Recall: 0.0220 Porcentaje de anomalías detectadas: 2.20% Transformador 7 Matriz de Confusión: [[ 43 457] [ 16 1960]] Precisión: 0.7288 Recall: 0.0860 Porcentaje de anomalías detectadas: 8.60% Transformador 8 Matriz de Confusión: [[ 12 488] [ 17 1959]] Precisión: 0.4138 Recall: 0.0240 Porcentaje de anomalías detectadas: 2.40% Transformador 9 Matriz de Confusión: [[ 9 491] [ 23 1953]] Precisión: 0.2812 Recall: 0.0180 Porcentaje de anomalías detectadas: 1.80% Transformador 10 Matriz de Confusión: [[ 14 486] [ 23 1953]] Precisión: 0.3784 Recall: 0.0280 Porcentaje de anomalías detectadas: 2.80% Transformador 11 Matriz de Confusión: [[ 8 492] [ 23 1953]] Precisión: 0.2581 Recall: 0.0160 Porcentaje de anomalías detectadas: 1.60% Transformador 12 Matriz de Confusión: [[ 34 466] [ 13 1963]] Precisión: 0.7234 Recall: 0.0680 Porcentaje de anomalías detectadas: 6.80% Transformador 13 Matriz de Confusión: [[ 10 490] [ 18 1958]] Precisión: 0.3571 Recall: 0.0200 Porcentaje de anomalías detectadas: 2.00% Transformador 14 Matriz de Confusión: [[ 15 485] [ 19 1957]] Precisión: 0.4412 Recall: 0.0300 Porcentaje de anomalías detectadas: 3.00% Transformador 15 Matriz de Confusión: [[ 31 469] [ 17 1959]] Precisión: 0.6458 Recall: 0.0620 Porcentaje de anomalías detectadas: 6.20% Transformador 16 Matriz de Confusión: [[ 15 485] [ 19 1957]] Precisión: 0.4412 Recall: 0.0300 Porcentaje de anomalías detectadas: 3.00% Transformador 17 Matriz de Confusión: [[ 47 453] [ 21 1955]] Precisión: 0.6912 Recall: 0.0940 Porcentaje de anomalías detectadas: 9.40% === Resultados usando factor de ruido: 0.15 === Transformador 1 Matriz de Confusión: [[ 20 480] [ 18 1958]] Precisión: 0.5263 Recall: 0.0400 Porcentaje de anomalías detectadas: 4.00% Transformador 2 Matriz de Confusión: [[ 21 479] [ 22 1954]] Precisión: 0.4884 Recall: 0.0420 Porcentaje de anomalías detectadas: 4.20% Transformador 3 Matriz de Confusión: [[ 22 478] [ 22 1954]] Precisión: 0.5000 Recall: 0.0440 Porcentaje de anomalías detectadas: 4.40% Transformador 4 Matriz de Confusión: [[ 33 467] [ 21 1955]] Precisión: 0.6111 Recall: 0.0660 Porcentaje de anomalías detectadas: 6.60% Transformador 5 Matriz de Confusión: [[ 93 407] [ 22 1954]] Precisión: 0.8087 Recall: 0.1860 Porcentaje de anomalías detectadas: 18.60% Transformador 6 Matriz de Confusión: [[ 22 478] [ 19 1957]] Precisión: 0.5366 Recall: 0.0440 Porcentaje de anomalías detectadas: 4.40% Transformador 7 Matriz de Confusión: [[ 52 448] [ 16 1960]] Precisión: 0.7647 Recall: 0.1040 Porcentaje de anomalías detectadas: 10.40% Transformador 8 Matriz de Confusión: [[ 18 482] [ 17 1959]] Precisión: 0.5143 Recall: 0.0360 Porcentaje de anomalías detectadas: 3.60% Transformador 9 Matriz de Confusión: [[ 15 485] [ 23 1953]] Precisión: 0.3947 Recall: 0.0300 Porcentaje de anomalías detectadas: 3.00% Transformador 10 Matriz de Confusión: [[ 14 486] [ 23 1953]] Precisión: 0.3784 Recall: 0.0280 Porcentaje de anomalías detectadas: 2.80% Transformador 11 Matriz de Confusión: [[ 8 492] [ 23 1953]] Precisión: 0.2581 Recall: 0.0160 Porcentaje de anomalías detectadas: 1.60% Transformador 12 Matriz de Confusión: [[ 41 459] [ 13 1963]] Precisión: 0.7593 Recall: 0.0820 Porcentaje de anomalías detectadas: 8.20% Transformador 13 Matriz de Confusión: [[ 26 474] [ 18 1958]] Precisión: 0.5909 Recall: 0.0520 Porcentaje de anomalías detectadas: 5.20% Transformador 14 Matriz de Confusión: [[ 32 468] [ 19 1957]] Precisión: 0.6275 Recall: 0.0640 Porcentaje de anomalías detectadas: 6.40% Transformador 15 Matriz de Confusión: [[ 34 466] [ 17 1959]] Precisión: 0.6667 Recall: 0.0680 Porcentaje de anomalías detectadas: 6.80% Transformador 16 Matriz de Confusión: [[ 22 478] [ 19 1957]] Precisión: 0.5366 Recall: 0.0440 Porcentaje de anomalías detectadas: 4.40% Transformador 17 Matriz de Confusión: [[ 73 427] [ 21 1955]] Precisión: 0.7766 Recall: 0.1460 Porcentaje de anomalías detectadas: 14.60% === Resultados usando factor de ruido: 0.2 === Transformador 1 Matriz de Confusión: [[ 38 462] [ 18 1958]] Precisión: 0.6786 Recall: 0.0760 Porcentaje de anomalías detectadas: 7.60% Transformador 2 Matriz de Confusión: [[ 22 478] [ 22 1954]] Precisión: 0.5000 Recall: 0.0440 Porcentaje de anomalías detectadas: 4.40% Transformador 3 Matriz de Confusión: [[ 22 478] [ 22 1954]] Precisión: 0.5000 Recall: 0.0440 Porcentaje de anomalías detectadas: 4.40% Transformador 4 Matriz de Confusión: [[ 46 454] [ 21 1955]] Precisión: 0.6866 Recall: 0.0920 Porcentaje de anomalías detectadas: 9.20% Transformador 5 Matriz de Confusión: [[ 147 353] [ 22 1954]] Precisión: 0.8698 Recall: 0.2940 Porcentaje de anomalías detectadas: 29.40% Transformador 6 Matriz de Confusión: [[ 29 471] [ 19 1957]] Precisión: 0.6042 Recall: 0.0580 Porcentaje de anomalías detectadas: 5.80% Transformador 7 Matriz de Confusión: [[ 92 408] [ 16 1960]] Precisión: 0.8519 Recall: 0.1840 Porcentaje de anomalías detectadas: 18.40% Transformador 8 Matriz de Confusión: [[ 24 476] [ 17 1959]] Precisión: 0.5854 Recall: 0.0480 Porcentaje de anomalías detectadas: 4.80% Transformador 9 Matriz de Confusión: [[ 18 482] [ 23 1953]] Precisión: 0.4390 Recall: 0.0360 Porcentaje de anomalías detectadas: 3.60% Transformador 10 Matriz de Confusión: [[ 18 482] [ 23 1953]] Precisión: 0.4390 Recall: 0.0360 Porcentaje de anomalías detectadas: 3.60% Transformador 11 Matriz de Confusión: [[ 17 483] [ 23 1953]] Precisión: 0.4250 Recall: 0.0340 Porcentaje de anomalías detectadas: 3.40% Transformador 12 Matriz de Confusión: [[ 73 427] [ 13 1963]] Precisión: 0.8488 Recall: 0.1460 Porcentaje de anomalías detectadas: 14.60% Transformador 13 Matriz de Confusión: [[ 49 451] [ 18 1958]] Precisión: 0.7313 Recall: 0.0980 Porcentaje de anomalías detectadas: 9.80% Transformador 14 Matriz de Confusión: [[ 37 463] [ 19 1957]] Precisión: 0.6607 Recall: 0.0740 Porcentaje de anomalías detectadas: 7.40% Transformador 15 Matriz de Confusión: [[ 54 446] [ 17 1959]] Precisión: 0.7606 Recall: 0.1080 Porcentaje de anomalías detectadas: 10.80% Transformador 16 Matriz de Confusión: [[ 27 473] [ 19 1957]] Precisión: 0.5870 Recall: 0.0540 Porcentaje de anomalías detectadas: 5.40% Transformador 17 Matriz de Confusión: [[ 99 401] [ 21 1955]] Precisión: 0.8250 Recall: 0.1980 Porcentaje de anomalías detectadas: 19.80% === Resultados usando factor de ruido: 0.25 === Transformador 1 Matriz de Confusión: [[ 36 464] [ 18 1958]] Precisión: 0.6667 Recall: 0.0720 Porcentaje de anomalías detectadas: 7.20% Transformador 2 Matriz de Confusión: [[ 32 468] [ 22 1954]] Precisión: 0.5926 Recall: 0.0640 Porcentaje de anomalías detectadas: 6.40% Transformador 3 Matriz de Confusión: [[ 31 469] [ 22 1954]] Precisión: 0.5849 Recall: 0.0620 Porcentaje de anomalías detectadas: 6.20% Transformador 4 Matriz de Confusión: [[ 70 430] [ 21 1955]] Precisión: 0.7692 Recall: 0.1400 Porcentaje de anomalías detectadas: 14.00% Transformador 5 Matriz de Confusión: [[ 227 273] [ 22 1954]] Precisión: 0.9116 Recall: 0.4540 Porcentaje de anomalías detectadas: 45.40% Transformador 6 Matriz de Confusión: [[ 34 466] [ 19 1957]] Precisión: 0.6415 Recall: 0.0680 Porcentaje de anomalías detectadas: 6.80% Transformador 7 Matriz de Confusión: [[ 115 385] [ 16 1960]] Precisión: 0.8779 Recall: 0.2300 Porcentaje de anomalías detectadas: 23.00% Transformador 8 Matriz de Confusión: [[ 34 466] [ 17 1959]] Precisión: 0.6667 Recall: 0.0680 Porcentaje de anomalías detectadas: 6.80% Transformador 9 Matriz de Confusión: [[ 28 472] [ 23 1953]] Precisión: 0.5490 Recall: 0.0560 Porcentaje de anomalías detectadas: 5.60% Transformador 10 Matriz de Confusión: [[ 19 481] [ 23 1953]] Precisión: 0.4524 Recall: 0.0380 Porcentaje de anomalías detectadas: 3.80% Transformador 11 Matriz de Confusión: [[ 21 479] [ 23 1953]] Precisión: 0.4773 Recall: 0.0420 Porcentaje de anomalías detectadas: 4.20% Transformador 12 Matriz de Confusión: [[ 80 420] [ 13 1963]] Precisión: 0.8602 Recall: 0.1600 Porcentaje de anomalías detectadas: 16.00% Transformador 13 Matriz de Confusión: [[ 71 429] [ 18 1958]] Precisión: 0.7978 Recall: 0.1420 Porcentaje de anomalías detectadas: 14.20% Transformador 14 Matriz de Confusión: [[ 45 455] [ 19 1957]] Precisión: 0.7031 Recall: 0.0900 Porcentaje de anomalías detectadas: 9.00% Transformador 15 Matriz de Confusión: [[ 66 434] [ 17 1959]] Precisión: 0.7952 Recall: 0.1320 Porcentaje de anomalías detectadas: 13.20% Transformador 16 Matriz de Confusión: [[ 26 474] [ 19 1957]] Precisión: 0.5778 Recall: 0.0520 Porcentaje de anomalías detectadas: 5.20% Transformador 17 Matriz de Confusión: [[ 126 374] [ 21 1955]] Precisión: 0.8571 Recall: 0.2520 Porcentaje de anomalías detectadas: 25.20% === Resultados usando factor de ruido: 0.3 === Transformador 1 Matriz de Confusión: [[ 46 454] [ 18 1958]] Precisión: 0.7188 Recall: 0.0920 Porcentaje de anomalías detectadas: 9.20% Transformador 2 Matriz de Confusión: [[ 41 459] [ 22 1954]] Precisión: 0.6508 Recall: 0.0820 Porcentaje de anomalías detectadas: 8.20% Transformador 3 Matriz de Confusión: [[ 36 464] [ 22 1954]] Precisión: 0.6207 Recall: 0.0720 Porcentaje de anomalías detectadas: 7.20% Transformador 4 Matriz de Confusión: [[ 74 426] [ 21 1955]] Precisión: 0.7789 Recall: 0.1480 Porcentaje de anomalías detectadas: 14.80% Transformador 5 Matriz de Confusión: [[ 286 214] [ 22 1954]] Precisión: 0.9286 Recall: 0.5720 Porcentaje de anomalías detectadas: 57.20% Transformador 6 Matriz de Confusión: [[ 44 456] [ 19 1957]] Precisión: 0.6984 Recall: 0.0880 Porcentaje de anomalías detectadas: 8.80% Transformador 7 Matriz de Confusión: [[ 177 323] [ 16 1960]] Precisión: 0.9171 Recall: 0.3540 Porcentaje de anomalías detectadas: 35.40% Transformador 8 Matriz de Confusión: [[ 35 465] [ 17 1959]] Precisión: 0.6731 Recall: 0.0700 Porcentaje de anomalías detectadas: 7.00% Transformador 9 Matriz de Confusión: [[ 36 464] [ 23 1953]] Precisión: 0.6102 Recall: 0.0720 Porcentaje de anomalías detectadas: 7.20% Transformador 10 Matriz de Confusión: [[ 24 476] [ 23 1953]] Precisión: 0.5106 Recall: 0.0480 Porcentaje de anomalías detectadas: 4.80% Transformador 11 Matriz de Confusión: [[ 19 481] [ 23 1953]] Precisión: 0.4524 Recall: 0.0380 Porcentaje de anomalías detectadas: 3.80% Transformador 12 Matriz de Confusión: [[ 125 375] [ 13 1963]] Precisión: 0.9058 Recall: 0.2500 Porcentaje de anomalías detectadas: 25.00% Transformador 13 Matriz de Confusión: [[ 107 393] [ 18 1958]] Precisión: 0.8560 Recall: 0.2140 Porcentaje de anomalías detectadas: 21.40% Transformador 14 Matriz de Confusión: [[ 62 438] [ 19 1957]] Precisión: 0.7654 Recall: 0.1240 Porcentaje de anomalías detectadas: 12.40% Transformador 15 Matriz de Confusión: [[ 82 418] [ 17 1959]] Precisión: 0.8283 Recall: 0.1640 Porcentaje de anomalías detectadas: 16.40% Transformador 16 Matriz de Confusión: [[ 47 453] [ 19 1957]] Precisión: 0.7121 Recall: 0.0940 Porcentaje de anomalías detectadas: 9.40% Transformador 17 Matriz de Confusión: [[ 178 322] [ 21 1955]] Precisión: 0.8945 Recall: 0.3560 Porcentaje de anomalías detectadas: 35.60% === Resultados usando factor de ruido: 0.35 === Transformador 1 Matriz de Confusión: [[ 52 448] [ 18 1958]] Precisión: 0.7429 Recall: 0.1040 Porcentaje de anomalías detectadas: 10.40% Transformador 2 Matriz de Confusión: [[ 67 433] [ 22 1954]] Precisión: 0.7528 Recall: 0.1340 Porcentaje de anomalías detectadas: 13.40% Transformador 3 Matriz de Confusión: [[ 49 451] [ 22 1954]] Precisión: 0.6901 Recall: 0.0980 Porcentaje de anomalías detectadas: 9.80% Transformador 4 Matriz de Confusión: [[ 96 404] [ 21 1955]] Precisión: 0.8205 Recall: 0.1920 Porcentaje de anomalías detectadas: 19.20% Transformador 5 Matriz de Confusión: [[ 335 165] [ 22 1954]] Precisión: 0.9384 Recall: 0.6700 Porcentaje de anomalías detectadas: 67.00% Transformador 6 Matriz de Confusión: [[ 49 451] [ 19 1957]] Precisión: 0.7206 Recall: 0.0980 Porcentaje de anomalías detectadas: 9.80% Transformador 7 Matriz de Confusión: [[ 204 296] [ 16 1960]] Precisión: 0.9273 Recall: 0.4080 Porcentaje de anomalías detectadas: 40.80% Transformador 8 Matriz de Confusión: [[ 46 454] [ 17 1959]] Precisión: 0.7302 Recall: 0.0920 Porcentaje de anomalías detectadas: 9.20% Transformador 9 Matriz de Confusión: [[ 42 458] [ 23 1953]] Precisión: 0.6462 Recall: 0.0840 Porcentaje de anomalías detectadas: 8.40% Transformador 10 Matriz de Confusión: [[ 30 470] [ 23 1953]] Precisión: 0.5660 Recall: 0.0600 Porcentaje de anomalías detectadas: 6.00% Transformador 11 Matriz de Confusión: [[ 32 468] [ 23 1953]] Precisión: 0.5818 Recall: 0.0640 Porcentaje de anomalías detectadas: 6.40% Transformador 12 Matriz de Confusión: [[ 116 384] [ 13 1963]] Precisión: 0.8992 Recall: 0.2320 Porcentaje de anomalías detectadas: 23.20% Transformador 13 Matriz de Confusión: [[ 134 366] [ 18 1958]] Precisión: 0.8816 Recall: 0.2680 Porcentaje de anomalías detectadas: 26.80% Transformador 14 Matriz de Confusión: [[ 66 434] [ 19 1957]] Precisión: 0.7765 Recall: 0.1320 Porcentaje de anomalías detectadas: 13.20% Transformador 15 Matriz de Confusión: [[ 112 388] [ 17 1959]] Precisión: 0.8682 Recall: 0.2240 Porcentaje de anomalías detectadas: 22.40% Transformador 16 Matriz de Confusión: [[ 56 444] [ 19 1957]] Precisión: 0.7467 Recall: 0.1120 Porcentaje de anomalías detectadas: 11.20% Transformador 17 Matriz de Confusión: [[ 186 314] [ 21 1955]] Precisión: 0.8986 Recall: 0.3720 Porcentaje de anomalías detectadas: 37.20% === Resultados usando factor de ruido: 0.4 === Transformador 1 Matriz de Confusión: [[ 79 421] [ 18 1958]] Precisión: 0.8144 Recall: 0.1580 Porcentaje de anomalías detectadas: 15.80% Transformador 2 Matriz de Confusión: [[ 82 418] [ 22 1954]] Precisión: 0.7885 Recall: 0.1640 Porcentaje de anomalías detectadas: 16.40% Transformador 3 Matriz de Confusión: [[ 51 449] [ 22 1954]] Precisión: 0.6986 Recall: 0.1020 Porcentaje de anomalías detectadas: 10.20% Transformador 4 Matriz de Confusión: [[ 98 402] [ 21 1955]] Precisión: 0.8235 Recall: 0.1960 Porcentaje de anomalías detectadas: 19.60% Transformador 5 Matriz de Confusión: [[ 376 124] [ 22 1954]] Precisión: 0.9447 Recall: 0.7520 Porcentaje de anomalías detectadas: 75.20% Transformador 6 Matriz de Confusión: [[ 55 445] [ 19 1957]] Precisión: 0.7432 Recall: 0.1100 Porcentaje de anomalías detectadas: 11.00% Transformador 7 Matriz de Confusión: [[ 244 256] [ 16 1960]] Precisión: 0.9385 Recall: 0.4880 Porcentaje de anomalías detectadas: 48.80% Transformador 8 Matriz de Confusión: [[ 42 458] [ 17 1959]] Precisión: 0.7119 Recall: 0.0840 Porcentaje de anomalías detectadas: 8.40% Transformador 9 Matriz de Confusión: [[ 48 452] [ 23 1953]] Precisión: 0.6761 Recall: 0.0960 Porcentaje de anomalías detectadas: 9.60% Transformador 10 Matriz de Confusión: [[ 33 467] [ 23 1953]] Precisión: 0.5893 Recall: 0.0660 Porcentaje de anomalías detectadas: 6.60% Transformador 11 Matriz de Confusión: [[ 36 464] [ 23 1953]] Precisión: 0.6102 Recall: 0.0720 Porcentaje de anomalías detectadas: 7.20% Transformador 12 Matriz de Confusión: [[ 149 351] [ 13 1963]] Precisión: 0.9198 Recall: 0.2980 Porcentaje de anomalías detectadas: 29.80% Transformador 13 Matriz de Confusión: [[ 170 330] [ 18 1958]] Precisión: 0.9043 Recall: 0.3400 Porcentaje de anomalías detectadas: 34.00% Transformador 14 Matriz de Confusión: [[ 68 432] [ 19 1957]] Precisión: 0.7816 Recall: 0.1360 Porcentaje de anomalías detectadas: 13.60% Transformador 15 Matriz de Confusión: [[ 127 373] [ 17 1959]] Precisión: 0.8819 Recall: 0.2540 Porcentaje de anomalías detectadas: 25.40% Transformador 16 Matriz de Confusión: [[ 62 438] [ 19 1957]] Precisión: 0.7654 Recall: 0.1240 Porcentaje de anomalías detectadas: 12.40% Transformador 17 Matriz de Confusión: [[ 198 302] [ 21 1955]] Precisión: 0.9041 Recall: 0.3960 Porcentaje de anomalías detectadas: 39.60% === Resultados usando factor de ruido: 0.45 === Transformador 1 Matriz de Confusión: [[ 90 410] [ 18 1958]] Precisión: 0.8333 Recall: 0.1800 Porcentaje de anomalías detectadas: 18.00% Transformador 2 Matriz de Confusión: [[ 116 384] [ 22 1954]] Precisión: 0.8406 Recall: 0.2320 Porcentaje de anomalías detectadas: 23.20% Transformador 3 Matriz de Confusión: [[ 73 427] [ 22 1954]] Precisión: 0.7684 Recall: 0.1460 Porcentaje de anomalías detectadas: 14.60% Transformador 4 Matriz de Confusión: [[ 115 385] [ 21 1955]] Precisión: 0.8456 Recall: 0.2300 Porcentaje de anomalías detectadas: 23.00% Transformador 5 Matriz de Confusión: [[ 408 92] [ 22 1954]] Precisión: 0.9488 Recall: 0.8160 Porcentaje de anomalías detectadas: 81.60% Transformador 6 Matriz de Confusión: [[ 64 436] [ 19 1957]] Precisión: 0.7711 Recall: 0.1280 Porcentaje de anomalías detectadas: 12.80% Transformador 7 Matriz de Confusión: [[ 285 215] [ 16 1960]] Precisión: 0.9468 Recall: 0.5700 Porcentaje de anomalías detectadas: 57.00% Transformador 8 Matriz de Confusión: [[ 51 449] [ 17 1959]] Precisión: 0.7500 Recall: 0.1020 Porcentaje de anomalías detectadas: 10.20% Transformador 9 Matriz de Confusión: [[ 60 440] [ 23 1953]] Precisión: 0.7229 Recall: 0.1200 Porcentaje de anomalías detectadas: 12.00% Transformador 10 Matriz de Confusión: [[ 49 451] [ 23 1953]] Precisión: 0.6806 Recall: 0.0980 Porcentaje de anomalías detectadas: 9.80% Transformador 11 Matriz de Confusión: [[ 49 451] [ 23 1953]] Precisión: 0.6806 Recall: 0.0980 Porcentaje de anomalías detectadas: 9.80% Transformador 12 Matriz de Confusión: [[ 171 329] [ 13 1963]] Precisión: 0.9293 Recall: 0.3420 Porcentaje de anomalías detectadas: 34.20% Transformador 13 Matriz de Confusión: [[ 204 296] [ 18 1958]] Precisión: 0.9189 Recall: 0.4080 Porcentaje de anomalías detectadas: 40.80% Transformador 14 Matriz de Confusión: [[ 76 424] [ 19 1957]] Precisión: 0.8000 Recall: 0.1520 Porcentaje de anomalías detectadas: 15.20% Transformador 15 Matriz de Confusión: [[ 154 346] [ 17 1959]] Precisión: 0.9006 Recall: 0.3080 Porcentaje de anomalías detectadas: 30.80% Transformador 16 Matriz de Confusión: [[ 83 417] [ 19 1957]] Precisión: 0.8137 Recall: 0.1660 Porcentaje de anomalías detectadas: 16.60% Transformador 17 Matriz de Confusión: [[ 209 291] [ 21 1955]] Precisión: 0.9087 Recall: 0.4180 Porcentaje de anomalías detectadas: 41.80% === Resultados usando factor de ruido: 0.5 === Transformador 1 Matriz de Confusión: [[ 109 391] [ 18 1958]] Precisión: 0.8583 Recall: 0.2180 Porcentaje de anomalías detectadas: 21.80% Transformador 2 Matriz de Confusión: [[ 155 345] [ 22 1954]] Precisión: 0.8757 Recall: 0.3100 Porcentaje de anomalías detectadas: 31.00% Transformador 3 Matriz de Confusión: [[ 101 399] [ 22 1954]] Precisión: 0.8211 Recall: 0.2020 Porcentaje de anomalías detectadas: 20.20% Transformador 4 Matriz de Confusión: [[ 118 382] [ 21 1955]] Precisión: 0.8489 Recall: 0.2360 Porcentaje de anomalías detectadas: 23.60% Transformador 5 Matriz de Confusión: [[ 438 62] [ 22 1954]] Precisión: 0.9522 Recall: 0.8760 Porcentaje de anomalías detectadas: 87.60% Transformador 6 Matriz de Confusión: [[ 88 412] [ 19 1957]] Precisión: 0.8224 Recall: 0.1760 Porcentaje de anomalías detectadas: 17.60% Transformador 7 Matriz de Confusión: [[ 323 177] [ 16 1960]] Precisión: 0.9528 Recall: 0.6460 Porcentaje de anomalías detectadas: 64.60% Transformador 8 Matriz de Confusión: [[ 58 442] [ 17 1959]] Precisión: 0.7733 Recall: 0.1160 Porcentaje de anomalías detectadas: 11.60% Transformador 9 Matriz de Confusión: [[ 71 429] [ 23 1953]] Precisión: 0.7553 Recall: 0.1420 Porcentaje de anomalías detectadas: 14.20% Transformador 10 Matriz de Confusión: [[ 47 453] [ 23 1953]] Precisión: 0.6714 Recall: 0.0940 Porcentaje de anomalías detectadas: 9.40% Transformador 11 Matriz de Confusión: [[ 55 445] [ 23 1953]] Precisión: 0.7051 Recall: 0.1100 Porcentaje de anomalías detectadas: 11.00% Transformador 12 Matriz de Confusión: [[ 150 350] [ 13 1963]] Precisión: 0.9202 Recall: 0.3000 Porcentaje de anomalías detectadas: 30.00% Transformador 13 Matriz de Confusión: [[ 219 281] [ 18 1958]] Precisión: 0.9241 Recall: 0.4380 Porcentaje de anomalías detectadas: 43.80% Transformador 14 Matriz de Confusión: [[ 68 432] [ 19 1957]] Precisión: 0.7816 Recall: 0.1360 Porcentaje de anomalías detectadas: 13.60% Transformador 15 Matriz de Confusión: [[ 166 334] [ 17 1959]] Precisión: 0.9071 Recall: 0.3320 Porcentaje de anomalías detectadas: 33.20% Transformador 16 Matriz de Confusión: [[ 91 409] [ 19 1957]] Precisión: 0.8273 Recall: 0.1820 Porcentaje de anomalías detectadas: 18.20% Transformador 17 Matriz de Confusión: [[ 222 278] [ 21 1955]] Precisión: 0.9136 Recall: 0.4440 Porcentaje de anomalías detectadas: 44.40%





Aquí tienes una interpretación de los resultados que has obtenido. Vamos a desglosarlo en varios puntos clave:

**1. Incremento del Factor de Ruido y Efecto en la Detección**

* Conforme aumentas el factor de ruido (de 0.05 a 0.5), el porcentaje de anomalías detectadas aumenta en general. Esto es porque, a mayor ruido, las modificaciones en los datos son más significativas, y el modelo de detección de anomalías (One-Class SVM) tiene más facilidad para identificar estos puntos alterados.
* Esto muestra que tu sistema es más efectivo para detectar anomalías cuando el nivel de ruido es alto, pero podría ser menos efectivo para detectar alteraciones pequeñas, lo cual es común en escenarios donde los ataques son sutiles.

**2. Porcentaje de Anomalías Detectadas por Transformador**

* Observando el porcentaje de detección para cada transformador, podemos ver que algunos transformadores tienen una mayor tasa de detección de anomalías en comparación con otros.
* Por ejemplo, en el nivel de ruido de 0.5:
  + El Transformador 5 detecta un 87.6% de las anomalías introducidas, lo cual es muy alto.
  + En cambio, el Transformador 10 detecta solo un 9.4% de las anomalías, lo que indica una variabilidad en la sensibilidad de cada transformador frente al mismo nivel de ruido.
* Esta variabilidad podría ser debido a diferencias en las características de los datos de cada transformador. Algunos transformadores pueden tener datos más susceptibles al ruido (como datos con una menor variabilidad natural), haciendo que las anomalías sean más detectables.

**3. Precisión y Recall en Diferentes Niveles de Ruido**

* A medida que aumenta el factor de ruido, tanto la precisión como el recall tienden a mejorar en la mayoría de los transformadores.
* **Precisión**: Indica la proporción de verdaderos positivos entre todas las detecciones de anomalías. Un valor de precisión alto sugiere que el modelo está haciendo predicciones más confiables, es decir, que los puntos clasificados como anomalías realmente lo son.
* **Recall**: Mide la proporción de verdaderas anomalías que el modelo es capaz de detectar. El aumento en el recall con mayores factores de ruido indica que el modelo está captando más anomalías conforme estas son más intensas (más ruido agregado).
* En transformadores donde la precisión y el recall son ambos altos para un nivel específico de ruido, el modelo es particularmente efectivo para ese transformador y nivel de ruido.

**4. Interpretación de la Matriz de Confusión**

* La matriz de confusión proporciona detalles sobre los verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos en cada transformador.
* Por ejemplo, en el transformador 5 con un factor de ruido de 0.5:
  + **Verdaderos Positivos (TP)**: 438, que indica los puntos alterados detectados correctamente.
  + **Falsos Negativos (FN)**: 62, que indica los puntos alterados que no fueron detectados como anomalías.
  + **Falsos Positivos (FP)**: 22, que son puntos normales clasificados incorrectamente como anomalías.
  + **Verdaderos Negativos (TN)**: 1954, que son puntos normales clasificados correctamente.
* Un alto valor de verdaderos positivos y un bajo número de falsos negativos es ideal, como se observa en el transformador 5 para este nivel de ruido.

**5. Consistencia en la Detección a Través de los Transformadores**

* No todos los transformadores muestran una detección consistente de anomalías al mismo nivel de ruido. Esto puede ser una limitación si buscas un sistema de detección uniforme.
* Algunos transformadores, como el 5 y el 7, tienden a tener altos porcentajes de detección y buena precisión y recall en niveles de ruido altos, mientras que otros transformadores, como el 10 y el 11, mantienen porcentajes de detección bajos en todos los niveles de ruido.
* Esta inconsistencia sugiere que el sistema puede necesitar ajustes específicos para cada transformador o que el One-Class SVM podría no ser la mejor opción para algunos de los transformadores debido a las características de sus datos.

**6. Conclusiones Generales**

* **Robustez del Modelo**: El modelo es bastante robusto frente a altos niveles de ruido, donde logra detectar la mayoría de las anomalías. Sin embargo, a niveles de ruido bajos (por debajo de 0.1), el sistema tiene dificultades para detectar las anomalías en la mayoría de los transformadores.
* **Transformadores Sensibles a Anomalías**: Transformadores como el 5 y el 7 parecen ser más sensibles a la detección de anomalías, mientras que transformadores como el 10 y el 11 son menos sensibles, lo cual podría requerir ajustes adicionales para mejorar la detección en estos últimos.
* **Sensibilidad al Ruido**: Al aumentar el ruido, el sistema mejora su desempeño en detección de anomalías, pero esto implica que, en un escenario real, el sistema puede fallar en detectar alteraciones más sutiles, lo que podría representar una vulnerabilidad.

**Recomendaciones**

* **Optimización de Umbrales**: Considerar ajustar los parámetros del One-Class SVM o explorar otros métodos de detección de anomalías para mejorar la sensibilidad en los transformadores menos robustos.
* **Estudios Adicionales por Transformador**: Analizar si cada transformador requiere una configuración específica de parámetros en el modelo de detección de anomalías.
* **Validación en Datos Reales**: Evaluar el modelo con datos reales o simulaciones que representen variaciones normales y alteraciones sutiles para asegurar que el sistema no dependa exclusivamente de niveles altos de ruido para una detección efectiva.

En resumen, estos resultados reflejan que tu sistema responde bien a niveles altos de ruido en la mayoría de los transformadores, pero puede requerir ajustes para mejorar su sensibilidad en transformadores menos sensibles y frente a anomalías sutiles.